Computers and Education

Towards an Interconnected Society

Manuel Ortega and José Bravo (Eds.)

2nd International Symposium on Computers in Education

Kluwer Academic Publishers

Computers and Education

Computers and Education
Towards an Interconnected Society

Edited by

Manuel Ortega

and

José Bravo

University of Castilla,
La Mancha, Ciudad Real, Spain

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-47533-2
Print ISBN: 0-7923-7188-7

Contents

SIIE'2000 Organisation

Programme Committee

President:
Dr. ANTONIO VAQUERO, U. Complutense (Spain)

Members:
Dr. ALFREDO FERNÁNDEZ VALMAYOR, U. Complutense (Spain)
Dr. ALVARO H. GALVIS, U. de los Andes (Colombia)
Dr. ANTONIO MOREIRA, U. Aveiro (Portugal)
Dr. BERT ZWANEVELD, O. U. of the Netherlands (The Netherlands)
Dr. CAROLYN DOWLING, Australian Catholic U. (Australia)
Dr. DUARTE COSTA PEREIRA, U. Porto (Portugal)
Dra. FELISA VERDEJO MAILLO, UNED (España)
Dra. ISABEL CHAGAS, U. Lisboa (Portugal)
Dra. ISABEL FERNÁNDEZ DE CASTRO, U. País Vasco (Spain)
Dra. JANNI NIELSEN, Copenhagen Business School (Denmark)
Dr. JESÚS LORÉS, U. Lleida (Spain)
Dr. JOHN TIFFIN, Victoria U. of Wellington (New Zeland)
Dr. JOSÉ BRAVO, UCLM (Spain)
Dr. JOSÉ LUIS RAMOS, U. Évora (Portugal)
Dr. JOSÉ MIGUEL ZAMARRO, U. Murcia (Spain)
Dr. JOSÉ M. TROYA, U. Málaga (Spain)
Dr. JULIÁN GUTIÉRREZ SERRANO, U. País Vasco (Spain)
Dr. KINSHUK, Massey U. (New Zeland)
Dr. LONE DIRCKINCK-HOLMFELD, Aalborg U. (Netherlands)
Dr. MANUEL PÉREZ COTA, U. Vigo (Spain)

Dra. MARIA JOSÉ MARCELINO, U. Coimbra (Portugal)
Dra. MARIA JOAO LOUREIRO, U. Aveiro (Portugal)
Dr. MARTÍN LLAMAS, U. Vigo (Spain)
Dr. MARTYN WILD, Edith Cowan U. (Australia)
Dr. NUNO GUIMARAES, U. Lisboa (Portugal)
Dr. PAOLO ROCCHI, IBM Education Research & Development (Italy)
Dr. PAULO DIAS, U. Minho (Portugal)
Dr. ROBERT LEWIS, U. de Lancaster (United Kingdom)
Dr. ROBERTO MORIYÓN, U. Autónoma Madrid (Spain)
Dra. ROSA MARIA BOTTINO, Ist. Matematica Applicata (Italy)
Dra. TERESA MENDES, U. Coimbra (Portugal)

Organisation Committee

President:
Dr. Manuel Ortega Cantero (UCLM-CHICO)

Members:
Dr. Antonio José Mendes (U. Coimbra)
Dr. Baltasar Fernández (U. Complutense)
D. Miguel Angel Redondo (UCLM-CHICO)
D. Crescencio Bravo (UCLM-CHICO)
Dr. Miguel Lacruz (UCLM-CHICO)
D. Pedro Pablo Sánchez Villalón (UCLM-CHICO)

ADIE
http://chico.inf-cr.uclm.es/adie
e-mail: adie@inf-cr.uclm.es

The CHICO Team (UCLM)
http://chico.inf-cr.uclm.es

Desktop Editor:
Pedro P. Sánchez Villalón

English Editor:
Pedro P. Sánchez Villalón

Acknowledgements

The editors would like to acknowledge the Universidad de Castilla – La Mancha, its Rector, Prof. Luis Arroyo, the “Escuela Superior de Informática” and the Computer Science Department for sponsoring the Symposium.

Special thanks to Rosa Maria Bottino, Lalita Rajasingham, John Tiffin and Paulo Dias, invited lecturers, for their high quality contribution to the success of the Symposium.

We would also like to thank the “Junta de Comunidades de Castilla- La Mancha” and especially the town of Puertollano (Ciudad Real), and the Major Mr. Casimiro Sánchez for providing their unconditional support to this event.

Manuel Ortega
President of the Organisation Committee.

Preface

SIIE is an international forum of Spanish-speaking, Portuguese-speaking and English-speaking researchers devoted to investigate and implement the use of computers in education.

In 1999 the Symposium was held in Aveiro, Portugal. In the year 2000 it was celebrated in Puertollano, Spain. Other meetings preceded this Symposium, namely, the "Simpósio de Investigação e Desenvolvimento de Software Educativo" held in Lisbon, Coimbra and Evora, two Congresses held in Spain and organised by ADIE: Encuentro de Informática Educativa, in Madrid and the so successful ConieD'99 held in Puertollano in 1999. A collection of Conied'99 papers is also published in this collection with the title "Computers in Education in the 21st Century" (2000).

ADIE (Association for the Development of Computers in Education) undertook the organisation of the congresses and symposiums on Computers in Education. It is an association which organises meetings for researchers in Spain, Portugal and Latin America. ADIE publishes "Revista de Enseñanza y Tecnología" in Latin America quarterly.

The 2nd International Symposium on Computers in Education brought together international groups of researchers to share experiences, making it possible to establish international relations enabling the realisation of European and other international projects. 135 papers were proposed to the Symposium. 50 papers and 11 posters were accepted for presentation. This book is the publication of the three plenary lectures and a selection of the 28 best papers presented at SIIE'2000. These proceedings deal with the topics established for discussion at the Symposium, namely, Human-Computer Interaction, Collaborative Learning, Distance Learning, Artificial Intelligence in Education, Multimedia and Hypermedia in Education, Intelligent Tutorial Systems, Educational Software: creation, development and evaluation, Teacher Training in Communication and Information Technologies, Educational Applications for People with Special Needs.

Manuel Ortega and José Bravo

PLENARY LECTURES

PART I

Artificial Intelligence in the HyperClass: Design Issues

John Tiffin, Nobuyoshi Terashima and Lalita Rajasingham

1. INTRODUCTION

On 15 December 2000 a three-way link was successfully made in HyperReality between Waseda University in Japan, Victoria University of Wellington in New Zealand and the Queensland Open Learning Network in Australia. The avatar of Lalita Rajasingham, who was in New Zealand, handed a virtual CD-ROM to the avatar of Anne Gooley in Australia, who then handed it to the avatar of Professor Nobuyoshi Terashima in Japan, who showed them how to slot it into a virtual computer. Three people in different countries interacted with each other as though they were in the same classroom, yet at each of the three sites only one person was physically present and the other two were virtually present. This was the outcome of five years of collaborative research between the authors. In previous experiments they had managed to stage a HyperClass between Japan and New Zealand but this time they established the technical viability of a HyperClass between multiple centres. It means that classes can be held which link a group in a conventional classroom with other groups of teachers and students who may be physically located anywhere in the world. This is happening already on the Internet but the HyperClass introduces a new level of interaction.

Although at the moment the degree of interaction in the experiment is only possible with high level computing facilities (the experiment used SGI 02) current use of the Internet for educational purposes makes it possible to envisage this kind of development becoming available for education some time in the first decade of the new millennium. There is another dimension.

The HyperClass is an application of HyperReality, a technology which also provides a platform for interactive communication between human intelligence and artificial intelligence. The HyperClass will make possible the widespread application of AI in education.

Nobuyoshi Terashima lead the team at the Advanced Telecommunications Research (ATR) Laboratories in Japan which first developed HyperReality (HR). HR provides a space for the interaction of virtual reality and physical reality and human intelligence and artificial intelligence [1]. It makes possible a future where virtual people, virtual objects and virtual settings can coexist and interact with real people, real objects and real settings and for human intelligence to comingle with artificial intelligence in a way that appears seamless. Clearly the technology used in the experiment is not yet at this stage. At this point the technology is gearing toward a first application of HR on the Internet with the kind of PC that we can imagine being widely available in a three or four years time. However, the use of curved widescreen technology in the original experiments at ATR made it possible for real and virtual people to work together in a perceptually seamless environment. We can imagine that, with the trend toward large flat visual display units, a more immersive form of HR will emerge in our living rooms and classrooms. In the long run HR is seen as a datasuit application which, like mobile phones, will be available almost anywhere.

John Tiffin and Lalita Rajasingham of Victoria University of Wellington have for the last 15 years been engaged in research into the design of virtual schools and virtual universities based on their concept of the virtual class [2]. Tiffin and Rajasingham define education from a neo-Vygotskian-perspective as a communication system that allows teachers to help learners to apply knowledge to problems. In conventional educational systems the means of communication for this are the transport systems that bring teachers and students together with their books, the classroom which allows face to face interaction and the textbooks, exercise books and whiteboard which house the knowledge and problems they are concerned with. This process can now be done by the Internet. A virtual class, therefore, means the interaction of teachers and students to apply knowledge to problems by telecommunications and computers.

In 1997 the three researchers came together to initiate a project to design a HyperClass [3]. Tiffin and Rajasingham had become interested in the relationship between conventional classes and virtual classes and Terashima could see in the virtual class a key application for HyperReality. The process in a virtual class is the same as in a conventional class but in one case it takes place between bits of information and in the other between atoms. It therefore becomes possible to think in terms of HyperReality making possible a HyperClass. This can be defined as “interaction between a real class and a virtual class about the application of a domain of knowledge to a relevant class of problems conducted in HyperReality”. However, HR introduces a new dimension. We need to add: “which may be enabled by AI and between teachers and students who may have artificial intelligence.

The first experiments between Waseda University and Victoria University demonstrated the technical feasibility of a HyperClass [3]. Students and teachers in Waseda and Victoria have been able to commingle in the study of ancient Japanese cultural artifacts. To the people in New Zealand the Japanese participants appeared as virtual people (avatars). Similarly to the Japanese the New Zealanders appeared as avatars. The artifacts were virtual objects based on a collection at the Tokyo University of Fine Arts. Later experiments incorporated a virtual computer whose components could be assembled. This allowed a simple teaching process where someone in the role of a teacher in one country could assist someone in the role of a student in another country when they had difficulties with the problem of assembling the computer.

Thus far, therefore, the HyperClass project has been concerned with the first aspect of HyperReality: with interaction for purposive communication between physical and virtual realities. Now the design issues are shifting to the second basic aspect of HR: that it provides a platform for communication between human and artificial intelligence. How does one develop a pedagogical communication system for a HyperClass which involves the interaction of artificial and human intelligence and the possibility that an avatar can represent a human intelligence or an artificial intelligence? The authors begin with Tiffin and Rajasingham’s neo-Vygotskian model of education as a communication system. This sees four critical communicative components: teachers, learners, knowledge and problems. Each of these four factors can be regarded as a potential point of application for intelligent agents. There is also the fact that a purposive system such as a class must have an organising subsystem. This represents another area where intelligent agents can be used.

2. ORGANISING AGENTS

A conventional class is a living system designed for learning. As such it needs a control system to bring teachers and students together, along with textbooks and exercise books, at fixed times and in fixed places with the purpose of learning a set curriculum and having this tested.. Every school, college and university has an administrative system based on human intelligence located in offices. It is rapidly becoming a norm for offices to have computers and telecommunications and an Internet connection. Agents are now being designed for use in such computerised office systems to help organise simple but, for humans, time-consuming tasks, such as setting up meetings and planning and adjusting the flow of events. In a conventional school bells will ring to announce the start and end of classes and everyone has a timetable that tells them which class to go to and when. Getting people together in a HyperClass could still involve this, but it could also be facilitated by an agent advising participants when to go to which classes. If they were on-line it could simply switch them from whatever they were doing into the HyperClass.

Students get lost in the detail of what they are supposed to be doing and what they are actually doing, they become confused about where a particular class or test fits into their overall progression through a subject and how and where they are achieving or failing. They seek to make meaning about themselves as learners in what to administrators and teachers can seem irritating questions about their grades and how they did on individual questions and exercises. Such questions involve looking up routine data and would seem to be territory for simple organising agents.

3. KNOWLEDGE AGENTS

The idea of knowbots has been around for some time. Agents are now widely used to search the web for specific information. A shopping agent will not only find services and goods advertised on the web but can also compare prices and terms. The parallel concept, the knowledge agent would find and evaluate the knowledge a learner or teacher wants at a particular moment.

Search engines in their various forms are seeking to become intelligent in an attempt to deal with the rapid growth and increasing complexity of the web. A HyperClass, however, while having a generic communicative function, is conceived as functioning in a coaction field that operates from a subject-specific data base [1]. The knowledge agent would, therefore, be specialist in searching, compiling, monitoring and evaluating knowledge only in the specific field of learning of a specific coaction field.

Given the growth of the web, knowledge agents will also need to be selective in terms of learners' needs. It would for example be possible to assess the language level of a particular knowledge base and relate this to the reading level of a student. A growing concern in the use of the web for education is with the reliability of the knowledge sources. A knowledge agent could form an estimate of the reliability of the knowledge through a count of references. Would this be any less discriminating than the personal evaluation of an individual teacher? Students already use the Internet to check the readings recommended by their teachers with those recommended by other teachers in other countries teaching similar courses.

4. PROBLEM AGENTS

Fundamental to the Vygotskian approach [4] is that knowledge has an application and education is about learning how to apply it to a domain of problems. What makes education different from other communication systems is that it is not just a flow of knowledge from teacher to learner. The learners learn to apply that knowledge and are tested to see how they cope with samples taken from the problem domain. Mathematics teachers set mathematical problems to see if students can apply the principles and theories which form the body of knowledge we call mathematics. In doing this millions of mathematics teachers around the world are marking billions of mathematical problems in the same way that mathematics teachers for thousands of years before them have done. The simpler the sum, the more repetitive the task. This is work for computers not humans and for many years now computerised devices have been available for providing problems that test elementary mathematical skills.

Most mathematical skills are closed. The problems have only one answer. Many of the basic skills in science subjects are also closed in the sense that students learn a fixed procedure which produces a fixed result. Various kinds of computerised testing systems have long been used to mark and evaluate performance in such subjects and these become increasingly effective and sophisticated. As life-long learners using a computer we are now subject to automatic checking of our spelling, punctuation and grammar and our PCs are becoming teaching machines.

What, however, of the subjects where the application of knowledge to problems is open to many interpretations? What about the levels of learning which centre on the questioning of the nature of knowledge and problems, where there are no perfect answers? Teachers involved in such subjects and levels of learning recognise that most students will at first follow fairly standard lines of enquiry until they engage the subject at a complex level that involves profound interaction with their teacher. Problem agents could sift through naïve attempts at problem solving and deal with Frequently Offered Answers and Frequently Asked Questions before human intelligence is involved. An agent attached to a particular problem domain would in time become expert in the different approaches to those problems. It would be able to recognise the different kinds of mistakes and provide appropriate cues, leaving new, unusual and difficult problems to human intelligence. As time goes by it may be that the agent would learn more and more of the possible interactions and the human would be left with the ever more exotic aspects of the problems.

An evaluation agent could collect and collate data across problems and students to make possible evaluation of the problems themselves or of learner behaviour in solving the problems. It could help draw a distinction between whether it is the problem that is badly set or the student who is badly prepared.

5. TEACHING AGENTS

At the heart of the Vygotskian approach expressed in the Zone of Proximal Development (ZPD) is the idea that when a learner has difficulty in applying knowledge to a problem they will learn more effectively if they can turn to someone in the role of teacher who can help them [4]. Tiffin and Rajasingham [2] noted that human teachers cannot be endlessly available but a teaching agent could. They also suggested that the help could be based on instructional hierarchies.

The structure of a problem [5] is not necessarily the structure of the knowledge needed to resolve it, nor are the structures of problems and of knowledge the same as the structures involved in the design of instruction. In instructional design the ability to apply knowledge to a category of problems is regarded as a skill or competency. These can in turn be seen as having hierarchical structures in which simple skills are mastered before proceeding to more complex higher level skills [6]. Hierarchies can be designed and iteratively tested in pursuit of optimum learning sequences. Theoretically this provides a basic mechanism for diagnosing where a learner is having difficulty. It also suggests a function for an instructional design agent that can patiently prompt and evaluate student progress against the possible paths through an instructional hierarchy. However, possibly the main strength of such an agent lies in a latent ability to endlessly monitor and iteratively improve the design of instruction.

6. LEARNING AGENTS

Where the design of learning hierarchies seeks the optimum learning path for the average student, the field of individualised learning recognises that different learners have different styles of learning. The constructivist approach in education takes this a step further in that learners are seen as giving their own structure to what they learn. Tiffin and Rajasingham [2] proposed learning agents that adapt over time to the learning needs of individuals and can even become life-long learning assistants. In this case an agent is needed that learns the way an individual learner learns in order to suggest the best learning pathways for them and to help them mature their learning styles. In doing so the agent would need to relate to knowledge structures, problem structures and instructional designs. This brings us to a point where we need to look at the integration of the different agent functions described above.

7. THE JITAIT (JUST IN TIME ARTIFICIAL INTELLIGENT TUTOR)

The AI agent functions are described above as discrete processes and we can recognise that in some form most of them already exist. They are, in the main, activities which a human teacher performs as a routine part of their job. However, what makes the human teacher special is that they do so in an integrated intuitive manner as part of their ongoing communication with their students. Could an AI virtual teacher in the form of an avatar implement these different functions as though they were linked processes and thereby simulate the basic role of a teacher? Initially we can suppose that the functions may be programmed separately and accessed as needed by the learner. Would it be such a big step to have an avatar cued by the learner to switch between modes in a manner that could become increasingly seamless? In due course the functions themselves could be increasingly integrated. Is there then anything inherently impossible or unlikely in the scenario below?

> ‘He (*an avatar in she shape of a wise old owl*) had been with her since she was a child. Of course in those days he looked like a cartoon. He would roost somewhere up in the right hand corner of her Computer Generated Virtual Reality and any time she raised her hand to him he would blink and look at her and be ready to help. He knew exactly where she was in her studies and that she hated to be told how to do things, preferring to work them out for herself, but he was so good at suggesting things she could try when she was really stuck. She got her first datasuit when she was in high school and then he became like a real owl. When she wanted him all she had to do was look at him and he would fly over and perch on her shoulder. His gentle whisper of encouragement and advice helped her through the bad patch when her parents divorced. It was through his guidance that she became a teacher herself. He was with her all through her time in the Trans-Pacific Teleteaching Training College. By that time she knew he was in touch

with other virtual teachers. With them he organised the virtual classes and it was through those links that he was able to advise her on how to relate to the other students with all their different cultures. Funny the shapes their virtual teachers took. The Thai student brought an elephant along with him and there was the girl from Peru with her virtual teacher as a snake coiled around her neck.

Now that she was a teacher herself he was still there for her. He knew so much about her. All she had to do was whisper his name and she could feel his grip on the shoulder of her datasuit ready to listen to the problems she was up against and look for the knowledge she needed. Trouble was when she had to lift the Head-Mounted Display Unit off to deal with the real world...'

(Tiffin and Rajasingham 1995) [2]

8. CONCLUSION

At the moment HyperReality technology is in an experimental stage and the form it will take when it is vended as an infrastructure technology has yet to emerge. However, interaction between virtual and physical worlds has been with us at least as long as the written word. The telephone allows us to talk anywhere with virtual voices and at its most basic, HR is simply the ability to telephone the whole person, plus the piano and the cat. We begin to see how this could be, when we look at the current genre of virtual worlds on the Internet and the increasing realism of computer games. The first experiments with a HyperClass establish that it is technically feasible to conduct a conventional class in conjunction with a virtual class. The process makes anywhere anytime education possible and already such institutions as the Queensland Open Learning Network and the School of the Future at the University of Sao Paulo are taking an interest in the possibilities for global HyperSchools and HyperUniversities. However, what may be the critical issue for the future of education is that the introduction of a HyperClass will also mean the widescale introduction of AI to education. There are already many simple uses of AI agents that could be applied in HR to assist the teaching process. However, it is in the long term, as HR technology evolves and becomes seamless and it becomes difficult to tell whether a student or teacher is real or virtual, that the issue of whether it is driven by human or artificial intelligence may come to dominate our concern.

REFERENCES

1. Terashima, N. (1995) 'HyperReality', *Proc. International Conference on Recent Advance in Mechatronics.*
2. Tiffin, J and Rajasingham, L (1995) *In Search of the Virtual Class: Education in an Information Society,* London and New York, Routledge
3. Terashima, N., Tiffin, J. and Rajasingham, L. (1999) 'Experiments on a virtual space distance education System using objects of cultural heritage', *Proc IEEE Computer Society MultimediaSystems* 2: 153.
4. Vygotsky, L.S. (1978) *Mind in Society: The development of the Higher Psychological Processes,* Cambridge MA: Harvard University Press
5. Tiffin, J (1978) Problems in Instructional Television in Latin America *Revista de Tecnología Educativa* 4(2)
6. Gagne, R.M. (1970) *Conditions of Learning,* Holt, Reinhart and Winston Inc

Advanced Learning Environments:
Changed Views and Future Perspectives

Rosa Maria Bottino
Consiglio Nazionale delle Ricerche - Istituto per la Matematica Applicata
Via De Marini 6 – 16149 – Genova – Italy
Email: bottino@ima.ge. cnr. it

1. INTRODUCTION

In the course of time the introduction of information and communication technologies (represented here and onwards with the acronym ICT) in education has been studied according to a range of perspectives. The benefits and the drawbacks of using technology in the school setting have been widely debated within the educational communities. Reference has been made to paradigms influenced, on the one hand, by the new possibilities offered by the rapid development of technology and, on the other hand, by the evolution of the cognitive and educational theories (explicitly or implicitly referred).

In the course of time, though with different approaches, methods and instruments, the objectives of the introduction of ICT in education have been substantially the following:

– developing new capabilities in the students permitting their integration in a society which computers have deeply modified;
– designing and/or using ICT-based tools for transforming and improving the teaching of disciplinary content.

In this work I refer to this second objective and I consider the role that ICT-based tools can assume in mediating teaching and learning processes. More specifically, I consider ICT-based tools explicitly designed with educational purposes. I will not discuss the role, even though important, that the use of "general purpose" tools, such as spreadsheets or software systems for drawing, can assume in education.

I will start with a brief discussion on the main aspects of the evolution that, since the beginning of the 80s, the research studies on the design and use of educational ICT-based systems has undertaken. I will then try to point out how the implementation of innovative learning environments, based on advanced technology, is the result of the strict interrelation between educational and cognitive theories, technological opportunities, and teaching and learning needs. As a matter of fact, the introduction of a new technology can bring with it a potential for education only if educationalists confront themselves with the necessity to understand, not only the technical and management problems related to its use, but also how the new possibilities offered by technology can help in the overcoming of problems usually encountered in didactical practice. Hence, this will occur only if they consider how technology can influence or change the nature of the pedagogy which can be developed thanks to its mediation.

According to this framework, my work discusses some principles for the design of effective learning environments to be used at school.

As an example, I refer to an innovative ICT-based tool, which has been designed and implemented to support the development of abilities in arithmetic problem solving with students in compulsory schooling.

2. SOME ELEMENTS CHARACTERISING THE EVOLUTION OF ICT-BASED EDUCATIONAL SYSTEMS

In this section I briefly consider some of the main orientations which deserve being pointed out in the research studies performed on the design and use of computers in education. Necessarily my discussion is neither detailed nor complete. The aim is to demonstrate that educational computing is a multidisciplinary research field where it is necessary to co-ordinate contributions and competencies taken from different fields: technology, ergonomics, computer science, psychology, besides those related to the specific application area (e.g., didactics of a given discipline).

2.1 The transmission metaphor

The first ways in which the computer had been used for educational purposes were influenced by behaviourism that considered learning as an induction of a required behaviour according to the well-known model "stimulus-response". The reference (implicit or explicit) to this model has brought to the design of drill and practice programs that are mainly aimed to exercise the student in the development of specific, often quite limited, competencies and abilities [1]. These programs, in the course of time, have undergone evolutions from a computer science point of view: from the first systems with rigid interfaces where all the possible answers were pre-programmed, to systems where the use of Artificial Intelligence techniques and methods allowed the customisation of the interface, the type of proposed exercises and the feedback obtained. Drill and practice systems, even now, represent the most relevant part of the educational computer-based systems available on the market. They usually employ some form of questioning strategy and often use some gaming techniques for encouraging participation and motivation. They include only minimal content instruction and are frequently used to test the acquisition of a given ability or to provide additional exercises to students who show problems or delay in the acquisition of some of the skills proposed by the curriculum. Ordinarily these programs are not used during the normal class activity but for individual training during "ad hoc" hours or at home.

Tutoring systems, differently from drill and practice systems, include content instruction in a given topic. They are often based on an information processing approach to learning. That is to say, in their design importance is ascribed to factors such as reinforcing memorisation, presenting objectives, specifying prerequisites, eliciting and assessing performance. Presented questions require application of the concepts or rules covered in the instructional sequences. Feedback is often diagnostic by identifying processing errors and prompting remediation or recasting of the instruction [2]. These systems are often thought as "stand alone" systems, designed as a single learner's private tutor. Their use in classroom practice is limited since they often are perceived more as replacements of teachers than as tools to help them in their work. It can be observed that this kind of approach to the use of computers in education grounds also in the basis of some distance learning courses based on the web that are now progressively widespreading.

Educational advantages in the use of both drill and practice and tutorials programs have revealed themselves quite limited. Their utility has been underlined in specific cases such as, for example, the performance of remedial activities (i.e. the handling of written calculations algorithms in arithmetic, or, in the domain of language, the development of spelling and grammar abilities) or the instruction on specific topics (i.e. initial training in the use of a software product).

These systems do not substantially change the way in which the user interacts with a given knowledge and does not contribute to create new ways to give meaning to concepts related to it. The prevailing metaphor is that of the system as an environment where knowledge is transmitted in order to be acquired by the user.

This metaphor grounds also in the development of many educational hypermedia systems. The main difference is that, by interacting with them, the user begins to assume an active role since she/he has the possibility to explore the presented contents following her/his needs and preferences according to personal paths [3].

2.2 The learner-centred metaphor

The increasing interest on constructivist theories has deeply changed the reference paradigm within which the computer is considered for learning aims. The attention has been progressively centred on the internal aspects of the student, on her/his attitudes and behaviours and on the cognitive processes that are involved in the learning interaction with the computer [4].

One of the major forces which has driven change has been the assumption that meanings are lost if learning is simply seen as the transmission of information. This approach is suitably expressed in key words used by many authors to describe and frame their own work; expressions such as "learner-centred systems" and "problem-based learning" have become more and more frequent in the literature (see, also, Norman and Spohrer, [5]). These terms suggest the idea that learning improves if the learner is immersed in some topic and is motivated to seek new knowledge and acquire new abilities by solving problems that the teacher has selected and organised so that the student can tackle them in a “natural” way. Hence, learning is viewed as being based on an active exploration and personal construction, rather than on a transmissible model.

Microworlds are an example of systems that are designed according to this general framework. The notion of microworld has undertaken a deep evolution since when, for the first time, this term was used by Minsky and Papert in a MIT report [6]. Initially the term microworld was used within Artificial Intelligence research studies to describe a domain of objects and activities (circumscribed and quite simple) that can be represented as a computer program. This concept was used to study and implement automatic strategies for the solution of problems. Classical examples were the world of blocks and the chess game. Papert was the first who investigated the possibilities offered by microworlds to the educational field. He considered the microworld notion with a slight but important change with respect to the meaning adopted till then: the simple (and constrained) domain was intended as part of a knowledge domain meaningful from an epistemological point of view [7]. So the notion of microworld changed - from a notion useful to instruct the computer to automatically solve problems in circumscribed and constrained contexts - to a notion useful to design environments suited for the learning of a given knowledge.

Even if there is no standard definition of the term 'microworld', there is agreement among researchers on a number of characteristics which are usually considered necessary for qualifying a system as a microworld (see, for example, Laborde and Strasser, [8]). For example, microworlds should provide the user with a number of primitives (objects and functions) that can be combined in order to produce a desired effect (computational, graphical, etc.). They should embody an abstract domain described in a model, and offer a variety of ways to achieve a goal. Moreover, they should allow the direct manipulation of objects, etc.

A microworld is built up around a given knowledge domain which has to be explored interacting with the program. Hence, in the design of microworlds for educational aims, a crucial role is assumed by the objects that are made available to the user through the interface of the microworld. Papert defined them as transitional computational objects, that is, objects which are inbetween the concrete and directly manipulated, and the symbolic and the abstract.

Consequently, an increasing importance is ascribed to the epistemology at the basis of a microworld as a key factor to distinguish between potentially powerful environments and environments less appropriate for exploration. The exploration is necessarily constrained but in a way suited to favour learning. In the mathematical field, a well-known example of this type of computer-based system is Cabri Geometre which has been designed to develop capacities in the formulation of conjectures and proofs in Euclidean geometry [9].

In general, the attention was on individual behaviour and the objective was to design and analyse learning situations where knowledge could emerge from interaction between the student and the computer environment.

The "learner-centred" approach applies the principles at the basis of the "user-centred" approach to the educational field. The "user-centred" approach has been assumed as reference in human-computer interaction studies for the construction of modern interfaces [10]. At the basis of this approach is the user (the learner) who has to solve a given task through the interaction with the system and the exploitation of its functionalities. Thus, the interest is on the needs, the abilities and the previous experiences of the user (the learner). The designer should anticipate the mental model of the system the user will develop so as to build up an interface that is suited for the development of given tasks.

Even if the above described orientations have brought to the development of a number of projects that have produced significant results at the research level, it is nevertheless true that the high expectations regarding ICT-based tools potential to drive change and innovation at school remain largely unfulfilled (see, Andrews [11]; Bottino and Furinghetti [12]; Pelgrum [13]). One of the main reasons for this (disregarding factors related to hardware availability and management, and to the traditional resistance of both the school system and teachers themselves to change) is that technology has often been introduced as an addition on to an existing, unchanged classroom setting [14].

2.3 The participation metaphor

Many research studies reveal that it is pointless from a pedagogical point of view to make computers available at school if the educational strategies and activities the students engage in are not suitably revised.

This observation often arises from analyses of how ICT are normally used in current practice. Often, a technological tool is used for educational purposes on the assumption that somehow or other it will lead to an educational improvement simply because the tool itself is considered to be "good". Seen in this light, technological tools are appreciated if they are rich in features or have a pleasant interface; no regard is made as to whether the tool in question is conceptually complex, whether it entails lengthy training before it can be used effectively, or how the teacher's role or teaching methods and contents need to be redefined to accommodate its use in the classroom [15]. This simplistic approach usually generates initial enthusiasm for a system, followed by disillusionment. The problem is that software environments are often evaluated on the basis of very general, ill-defined expectations, resulting in a lack of understanding about the conditions under which the educational use of such tools might be meaningful.

In recent years, this issue has represented a major topic for discussion in the debate that researchers have been conducting in the domain of educational computing. At theoretical level, we have assisted to a progressive move from cognitive theories that emphasise individual thinkers and their isolated minds to theories that emphasise the social nature of cognition and meaning [16]. An increasing importance is ascribed to theories that highlight the importance of studying the relations among individuals, mediating tools, and the social group (reference can be made to theories such as Activity theory, Situated Action Models, Distributed Cognition. For a short account on these theories, see, for example, Nardi [17]). These theories suggest a reformulation of learning in which practice is not conceived as independent of learning and in which meaning is not conceived as separate from the practices and contexts in which it is negotiated.

Technology design and use are thus being progressively considered in relation with the whole teaching and learning activity and not merely with the development of specific abilities and/or the accomplishment of particular tasks. For example, Bellamy [18] observes how technologies must be designed to support not only students' learning activities but also teachers' activity, since it is only by understanding and designing for the whole education situation that effective and valuable changes can be brought about in the classroom.

Increasingly, technology is being studied in relation with long-term teaching and learning processes of the kind needed for the development of complex articulated knowledge (e.g., arithmetic problem solving, Newtonian principles in Physics, comprehension and communication in language, etc.). For the development of such abilities, the student-software unit of analysis does not suffice as it is necessary to consider the whole set of interactions established in a class over the course of time.

As the matter of fact the mediation offered by a given software to cognition, is not sufficient to explain the learning aspects related with motivation, with goals formation and with the attribution of a meaning to the whole activity which goes beyond the meaning of the single actions involved in the performance of a task. The analysis of these aspects requires looking at learning not only as an individual construction developed during the interaction with the computer but also as a social construction developed within the whole learning environment.

The concept of learning environment itself has undergone a deep transformation in the course of time and its evolution has substantially contribute to change the way in which the mediation offered by technology to educational processes is considered. At the beginning, as it is witnessed by the terminology frequently adopted in the literature, educational software applications are often referred to as learning environments, thus focusing attention on the fact that it was the software itself, through interaction with the student, that was to form the environment where learning can be developed. Progressively the term learning environment is applied to the learning situation as a whole where technology is ascribed an important mediating role but does not cover the whole environment.

Consequently there is an increasing interest in aspects related not only to software design but also in the definition of ways of use suitable for exploiting software features in order to accomplish meaningful teaching and learning activities.

This interest for the environmental aspects is not confined to research in educational computing but is also being expressed in research areas like social ergonomics and Human Computer Interaction (HCI) . Studies being conducted in these fields consider the environment in which a given tool is used as an integral part of the tool itself (see, for example, the works of Bevan and Macleod [19], and Suchman [20]).

This shifted paradigm has two different implications for the implementation of effective ICT-mediated learning environments. Technological tools influence and transform the activities performed with their mediation, but, on the other hand, practice can deeply influence the technology used. This is particularly true now, when technological progress is constantly opening up new opportunities (for elaboration, representation, communication, etc.) whose potential for educational purposes has yet to be fully exploited. In other words, according to this approach, the way in which technology can be used in social practice can prefigure new functions to be included in the technology. These new functions and opportunities can change the models of practice which have inspired the construction of the technology itself [21].

It can be noted that the development of new models of practice can prefigure new ways of using existing ICT-based tools that can change the role that such systems has previously had and, consequently, the mediation that they can offer to teaching and learning processes. For example, in appropriately designed didactic situations, a hypermedia system can be used to introduce students to activities of meta-cognition, such as a reflection on their ways of organising knowledge in a given field, instead of as a tool to deliver information on a given topic field (see, for example, the experience reported in Bottino et al. [22]).

2.4 Towards a new generation of open learning systems

Some observations on the design and use of ICT-based educational systems can be made to conclude the brief analysis performed in the preceding sections.

We can observe that, in general, school curricula subsume different types of teaching and learning methods, and so no single method or type of tool used can be the choice for all occasions. Moreover, within any learning domain, students' and teachers' needs evolve over the activities in which they are involved and the tools have to support this evolution.

The performed analysis prefigures the possibility to develop a new generation of open-learning systems which can be more appropriate to mediate the new ways of looking at teaching and learning processes that are now progressively affirming themselves. These systems should make available tools able to support not only the relationship of the student with the learning object but also all the relationships that are established between participants during a teaching and learning activity.

It is possible to delineate some general indications for the design and analysis of such systems. Of course, these indications are to be detailed and motivated according to the characteristics of the specific field of application and educational context considered.

In particular, in the design of ICT-based educational systems, the following issues assumes a crucial importance:

- The computational objects and interactivity that a system makes available to the user and their relationship with the cognitive processes involved in the acquisition of the knowledge for the learning of which the system has been realised.
- The tools offered to validate student's action and the support they offer to the evolution of student's knowledge.
- The tools offered to support the re-elaboration of personal experience and its sharing within the class.
- The tools offered to support the setting up of a social context able to assist students' performance and to favour the transformations of the object of the teaching and learning activity that are necessary for acquiring a mastery of a given knowledge domain.

Of course tools and features cannot by themselves guarantee learning. They have to be used in order to support the construction of activities in which learning could be the result of a social construction of meanings and of their justifications.

3. AN EXAMPLE OF AN OPEN LEARNING MULTI-ENVIRONMENT SYSTEM

In the following, I briefly describe a system which has been designed taking into consideration the general issues above described. This system is a multi-environment one that has been implemented to support the development of abilities in arithmetic problem solving with primary and lower secondary school students. The objective of the description is to give an example of how the general indications evidenced in the previous section can have a practical correspondence. The final aim is to show how a system of this kind can support the design of teaching and learning activities meaningful for the acquisition of a given knowledge.

ARI-LAB is a system that combines hypermedia and network communication technologies in order to support the teaching and learning activity in arithmetic problem solving.

The project of ARI-LAB is based, on the one hand, on research on hypermedia and communication systems and on the design and implementation of visual microworlds; on the other hand it takes into account the research in mathematics education with particular reference to studies on applied problem solving, on the role of visual representations in learning processes and on interactive learning.

The development of the ARI-LAB project has been an iterative process based on the wide experiments of the system we have performed on the long term in real classroom settings. Thus, the current version, now available on the market, is deeply changed from the first prototype implemented [23]. The design of ARI-LAB (which has included the design of possible contexts of use for the system) has been based on the model of human activity that Engestrom has built on Leont'ev's 'Activity Theory' [24]. This interprets human learning as a systemic formation, taking into account its tool-mediated, object-oriented nature and its grounding in communities of practice (see [25]).

3.1 A short overview of the system

ARI-LAB is an open-learning system including a structured and connected set of tools combining hypermedia and network communication technologies. The aim of the system is to support the construction of class teaching and learning activities in arithmetic problem solving (at primary and lower secondary school level).

The problem solving activity is carried out by exploiting the action, representation and communication possibilities made available by the different tools which have been integrated in the system. These are: tools to develop solution processes (microworlds); a tool to build and present solutions (solution sheet); a tool allowing communication between users (the communication environment); a tool to store all the actions the user performed in the microworlds and in the solution sheet while solving a problem (monitoring). Moreover, the teacher is granted access to yet another tool (the teacher environment) which enables him/her to plan the didactical activity and to configure the system according to educational goals and specific needs of the students involved.

3.2 Microworlds

Microworlds are the mediating tools for the construction of solution processes. Within microworlds, the user can create and manipulate computational objects to develop the solution strategy for a problem. The microworlds currently available are called: Coins, Calendar, Abacus, Number Building, Number Line, Simplified Spreadsheet, Histogram, and Art Bits.

The first two microworlds have been designed to support the construction of solution strategies for additive and multiplicative problems in fields that are related to students' real world experience (buying and selling, measure of time). Abacus, Number Building and Number Line microworlds have been designed to facilitate the development of capacities in numbers representation and counting. The Spreadsheet and Histogram microworlds are intended to support activities in fields such as that of real data handling and formulae manipulation. Art Bits is a microworld for drawing. In each microworld the user is provided with a visual language which allows him/her to perform different kind of actions in order to obtain a visual representation which can be useful for the solution of the problem at hand. For example, in the coin microworld, coins of the Italian, or Euro, currency system can be generated, moved on the screen, changed with others of the same value, etc. . Some actions are available in all microworlds, such as the opportunity to copy a visual representation in order to paste it in the solution sheet. Some microworlds make available functions for the validation of the counting process undertaken by the user. For example, in the coin microworld, it is possible to select a (previously generated) coin or group of coins and to hear the amount pronounced orally by means of a voice synthesiser incorporated in the system. This feedback can also be obtained in the form of a movie showing the amount in sign language so as to support the learning activity of deaf children. The same kind of feedback can be obtained for a representation accomplished in the abacus or in the number building microworlds.

3.3 Solution Sheet

The solution sheet is the environment where the solution to a problem is built up as a product to be shared with others (teachers, other students of the same class, etc).

In the solution sheet the user builds up his/her solution to the problem at hand by copying into this space the visual representations produced in the microworlds that s/he considered meaningful for problem solution. The user employs verbal language and arithmetic symbolism to comment on the visual representations copied and thus to explain the solution performed.

3.4 Monitoring

ARI-LAB features a tool that automatically records anything a pupil has done while solving a problem (every action performed in the microworlds and in the solution sheet). This tool, called 'monitoring', makes it possible, at any time, to view the whole sequence of steps performed as a sort of movie.

The tool transforms the resolution process undertaken in the microworlds into an object that can be used in the activity for different didactical aims. For example, it mediates the possibility to describe and interpret the solution process through verbal and symbolic language, even after some time has elapsed.

3.5 The Communication Tool

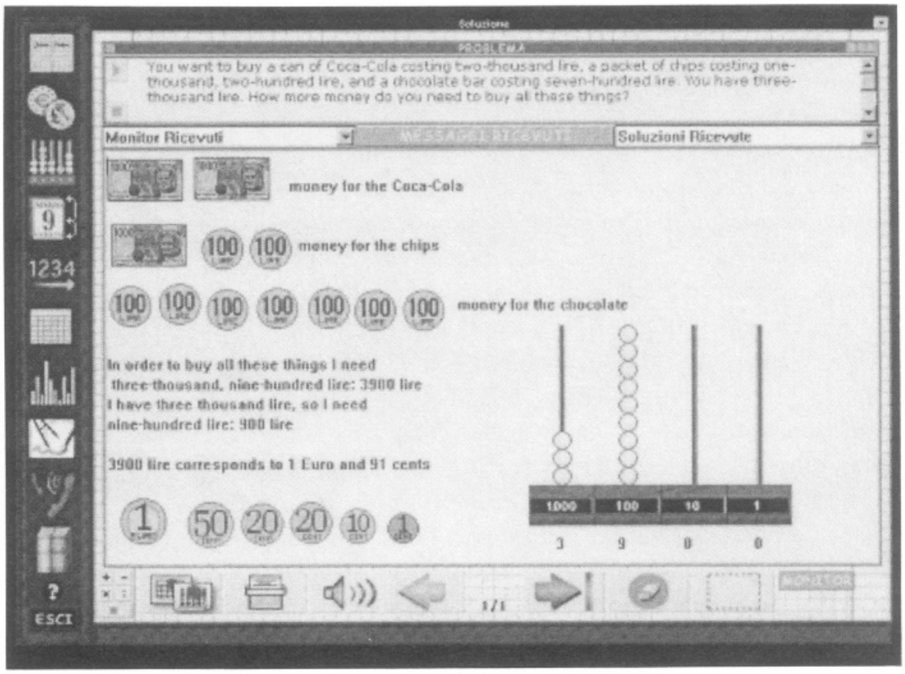
While solving a problem with ARI-LAB, it is possible at any moment to access the Communication tool, which allows the user to establish a connection with another user and share messages by means of a local network. The other user may either be a classmate or the teacher. If the two interlocutors are solving the same problem, it is also possible for either of them to send the other their solution (all the work performed till that moment in the solution sheet) and the related monitoring. In this way the receiver not only sees his/her classmate's solution to the problem, but also how that solution was reached.

It is worth noting that a solution (or solutions) received by a pupil from classmates cannot be copied into the pupil's solution sheet. Access is granted for the sake of analysis but, in order to use the received solutions in his/her solution sheet, the pupil has to reconstruct them through interaction with the microworlds.

3.6 Teacher's environment

The teacher is granted access to an environment which allows him/her to configure the system according to the specific needs of the students involved, e.g. to define and impart to the students a set of arithmetic problems to solve, to choose which microworlds should be accessible to the students, to make available or inhibit the validation functions (e.g. voice synthesiser or films in sign language). Hence, the system interface can be changed in accordance with the teacher's options.

Figure 1 shows the solution sheet produced by a user tackling the problem whose text is reported in the upper right-hand side of the figure. On the left-hand side there are the buttons which allow access to the microworlds and to the communication tool. For each problem, it is possible to access the entire dialogue conducted by the user with other students while solving the same problem ("messaggi ricevuti" button). Moreover, at any moment of the activity it is possible to select the monitoring function ("monitor") in order to see the whole sequence of actions the user has performed while solving that problem. It is also possible to access the solutions to the same problem received from other users ("soluzioni ricevute") and the related monitoring ("monitor ricevuti"). In this way, the whole individual and social activity conducted in ARI-LAB to solve a problem is incorporated and stored in the system.



You want to buy a can of Coca-Cola costing two-thousand lire, a packet of chips costing one-thousand, two-hundred lire, and a chocolate bar costing seven-hundred lire. You have three-thousand lire. How more money do you need to buy all these things?

In order to buy all these things I need
three-thousand, nine-hundred lire: 3900 lire
I have three thousand lire, so I need
nine-hundred lire: 900 lire

3900 lire corresponds to 1 Euro and 91 cents

Figure 1. Interface of the ARI-LAB Solution Sheet

3.7 Issues arising in the design and use of ARI-LAB

In the design of the system we start from the pedagogic consideration that primary school pupils (and often secondary school students) usually have a serious difficulty tackling arithmetic problem solving and teachers have trouble assisting them adequately. The school tradition relies on the early introduction of arithmetic symbols and written computation algorithms as the only way to describe the solution process and to obtain the result. This approach seldom works well, as is witnessed by the fact that too many students, when solving a problem, try to "guess" what operation is necessary: they are not able to give meaning to the arithmetical signs in relation with the situation described in the problem. The introduction of arithmetical symbols and written computation can be introduced fruitfully when children have already experienced the potentialities of a numbering system, and are able to enact informal strategies within concrete problem situations. Representation systems such as coins or calendar can act as mediators between the situation described in the problem and the mathematical ideas, relations and processes involved in the solution. In this way the mathematical ideas involved in problems of additive and multiplicative structure are made more accessible to pupils, by means of perceptive-type control, grounded in the children's extra-school experience.

The pedagogical objective for which the microworlds have been implemented is to offer students a space in which they can explore and manipulate graphical and computational objects designed to mediate the development of solution processes and thus the construction of meanings for arithmetic operations. The system supports the validation of specific actions or processes (e.g. counting, changing coins or balls in the abacus, etc.) offering a perceptive feedback (e.g. voice synthesis). The feedback obtained allows the user to progressively acquire competencies on specific aspects of the knowledge involved in the interaction with the microworlds and can support the development of didactical activities aimed to foster the acquisition of crucial capabilities such as, for example, the co-ordination of verbal, graphical and written representations of numbers.

Moreover tools are offered to support the re-elaboration of personal experience and its sharing within the class. The monitoring tool and the solution sheet support the user in expressing the resolution process as a product to be communicated to others. Moreover, monitoring, which is the possibility to view, in a sort of movie form, all the actions performed while solving a problem, can support the transformation of the solution procedure into an object which can be used as a basis for discussion in the social context of the class. This discussion may have different aims, such as the comparison of strategies, the analysis of the mathematical properties (and of their relationships) involved in the solution strategies undertaken, etc.

In designing ARI-LAB, an alternative architecture to the usual one present in traditional educational systems (student-computer) has been adopted. For this reason, a communication tool has been integrated into ARI-LAB so that users can engage in problem-solving within a social interaction process. The communication tool can change the form and status of an arithmetic problem, students' assumptions on how to solve it and the validation context in which the resolution process is set. Moreover, it can change the rules governing the behaviour of the participants in the activity, and the different roles the participants may take. One condition for this is that communication becomes an integral part of the problem solving activity and that students develop a synergy between communication and the other activities performed (building a representation, explaining a solution, examining a monitoring, analysing a received solution, etc.). The system supports this synergy by offering in a unique interface layout the different tools it is composed of so that the user can easily switch from one to the other at any time.

The ARI-LAB system has been experimented with widely and in depth in various school situations with children who are both normal and disabled (deaf). In particular, with one class, experiment was carried out for almost the whole cycle of primary school (from the second to the fifth class), integrating its use in the teaching of most of the mathematics programme. In all experiments carried out, the teachers of the classes involved were always present and participated actively both in setting up the teaching itineraries and in following the students' work on the computer. The research has had a very positive outcome in a sector which has always proved difficult for students and in which traditionally teachers find major problems about the approach to follow (see Bottino and Chiappini [25], and Bottino and Chiappini [26]).

4. CONCLUDING REMARKS

The ARI-LAB system is an example of a new generation of multimedia learning systems for school education. This new generation of systems is characterised by a strict integration of visualisation, communication and re-elaboration tools aimed to support both the creative exploration of problems and the representation and justification of knowledge. In ARI-LAB the multimedia paradigm assumes a new meaning: there is not only communication of knowledge but tools are provided to construct it by means of different ways of interaction. The user becomes an active subject not only when navigating through the system, but also when building new knowledge.

The new model of systems that ARI-LAB contributes to define is characterised by the possibility to be integrated in the school curricula in order to be used to develop not only single and specific abilities but concepts and understandings which requires a long period of time to be developed and learned (as it is the case of arithmetic problem solving). This can be done according with consistent pedagogical models.

Moreover, the integration in a unique environment of tools to support children with special needs (such as deaf children) introduces a new way of looking to special education (integrated in the activities of the entire classroom and not a-part from it).

The discussion carried out in this paper is grounded on the assumption that learning processes cannot be fully understood if only the individual learner is considered without taking into account the whole teaching and learning situation, that is to say, the relationships that are established between the actors involved (the student, the teacher, the other students) and the role played by mediating tools.

According to this view, the analysis of the way in which a ICT-based tool can affect learning is to be performed not only analysing its specific inner characteristics and how they affect the interaction with the individual user but also all the activity that is developed through its mediation in the context of use. From one hand, the didactical functions supported by the tool influence the way in which the teaching and learning activity can be carried out, but, on the other hand, these functions cannot be fully exploited if the whole learning environment is not considered.

REFERENCES

1. Reigeluth, C.M. (Ed.): 1987, Instructional theories in action: lessons illustrating selected theories and models, Hillsdale, NJ: Lawrence Erlbaum Associates.
2. Case, R., Bereiter, C.: 1984, From behaviorism to cognitive behaviorism: steps in the evolution of instructional design, Istructional Science, 13, 141-158.
3. Tomek, I., Khan, S., Muldner, T., Nassar, M, Novak, G., & Proszynski, P.: 1991, ‘Hypermedia- Introduction and Survey’, Journal of microcomputer applications, v.14, 63-103.
4. Brown J.S., Collins, A., Duguid, P.: 1989, Situated cognition and the culture of learning, Educational Researcher, 18, 32-42.
5. Norman, D. A., Spohrer, J.C.: 1996, Learner Centered Education, Communication of the ACM, 39/4, 24-27.
6. Lawer, R.W.: 1987, Learning environments: now, then and someday, in Lawer, R.W. & Yazdani, M. (eds.), Artificial Intelligence and Education, Vol. 1, Norwood, NJ:Ablex, 1-26.
7. Papert, S.: 1980, Mindstorms: children, computers, and powerful ideas, New York: Basic Book.
8. Laborde, J.M., Strasser, R.: 1990, ‘Cabri-Géomètre: a microworld of geometry for guided discovery learning’, ZDM, 90/5, 171-177.
9. Laborde, C.: 1993, The computer as part of the learning environment: the case of geometry, in Keitel, C. & Ruthven, K. (eds.), Learning from Computers: Mathematics Education and Technology, Nato Asi Series F, 121, Berlin: Springer Verlag, 48-67.
10. Norman, D. A., Draper, S.: 1986, User Centered System Design, Hillsdale JY: Lawrence Erlbaum Associates.
11. Andrews, P.: 1999, Some Institutional Influences on Secondary School Mathematics Teachers' Use of Computers, Education, and Information Technologies, 4/2, 113-128.
12. Bottino, R.M., Furinghetti, F.: 1998, The Computer In Mathematics Teaching: Scenes From The Classroom, Information and Communications Technologies in School Mathematics, in Tinsley, D., and Johnson, D.C. (eds.), London: Chapman & Hall, IFIP Series, Chapter 16, 131-139.

13. Pelgrum, W.J.:1996, The Educational Potential of New Information Technologies: Where are we now?, in Collis, B.A. (ed.), Children and Computers in School, Mahwah, NJ: Lawrence Erlbaum.
14. De Corte, E.: 1996, Changing Views of Computer Supported Learning Environments for the Acquisition of Knowledge and Thinking Skills, in Vosniadou S., De Corte E., Glaser R. and Mandl H. (eds.), International Perspectives on the Desing of Technology-Supported Learning Environments, Mahwah, USA: Lawrence Erlbaum Associates, 129-145.
15. Noss, R.: 1995, Thematic Chapter: Computers as Commodities, in diSessa A.A., Hoyles C. , Noss R. (eds): Computers and exploratory learning, Nato Asi Series F, Vol. 146, Berlin: Springer Verlag, 363-381.
16. Resnick, L.B.: 1987, Learning in school and out, Educational Researcher, 16, 13-20.
17. Nardi, B.A.: 1996, Studying context: a comparison of Activity Theory, Situated Action Models, and Distributed Cognition, in Nardi, B.A. (ed.), Context and Consciousness, Cambridge MA: The MIT press, 69-102.
18. Bellamy, R. K. E.: 1996, Designing educational technology, in Nardi, B.A (ed.) Context and Consciousness, Cambridge, MA: The MIT Press, 123-146.
19. Bevan, N., Macleod, M.: 1994: Usability measurement in context, Behaviour & Information Technology, 13/1 & 2, 132-145.
20. Suchman, L.A.: 1987, Plans and Situated Actions, Cambridge, MA: Cambridge University Press.
21. Pea, R.D.: 1987, "Cognitive Technologies for Mathematics Education in Schoenfeld, A.H., Cognitive Science and Mathematics Education, Hillsdale, NJ: Lawrence Erlbaum Associates, 89-122.
22. Bottino, R.M., Cutugno, P., Furinghetti, F.: 1998, Hypermedia as a means for learning and for thinking about learning, in Ottmann, T., and Tomek, I., (eds.) Proceedings of ED-MEDIA/ED-TELECOM 98, AACE, USA: Charlottesville, Vol.1, 144-149.
23. Bottino, R.M., Chiappini, G., Ferrari, P.L.: 1994, A hypermedia system for interactive problem solving in arithmetic, Journal of Educational Multimedia and Hypermedia, AACE, 3, 3/4, 1994, 307-326.
24. Engeström, Y.: 1991, Activity Theory and individual and social transformation, Activity Theory 7/8.
25. Bottino, R.M., Chiappini, G.: to appear, Advanced technology and learning environments.
26. Bottino, R.M., Chiappini, G.: 1997, La natura della mediazione offerta dai sistemi basati su micromondi all' apprendimento della matematica, Parte II: Analisi dei risultati delle sperimentazioni del sistema ARI-LAB, L'Insegnamento della Matematica e delle Scienze Integrate, Vol. 20A-B, 6,.792-838.

Learning Communities in the Web:
Concepts and strategies

Paulo Dias
Paulodias@nonio. uminho.pt
University of Minho, Portugal

1. INTRODUCTION

The emergence of on-line learning communities is supported by the growing culture of participation in the learning activities through the interactional process. Starting from this point, a learning community develops itself in a classroom or on the Web, when all the members of the group, including the teacher or the tutor, are deeply involved in the process of knowledge building, first as mutual engagement in the community formation and then on the development of their learning activities.

Learning communities work as cooperative and distributed systems, that raise from interaction and its effects through the communication among their members.

The engagement of members in a common learning goal, and the specific dimensions of communication in the process of knowledge building, concerning adjustment and the adaptation of individual representations among the members, are some of the aspects that sustain the organisational and learning activities in the community creation. This also draws the development of a culture of sharing and collaborative activities, which defines on-line learning communities. According to Rogers [1], the main characteristic of the responsibility of the shared learning, promotes the knowledge distribution among the members of the group, and the use of individual knowledge and competencies in the growth and definition of the learning paths of the community itself.

M. Ortega and J. Bravo (eds.), Computers and Education, 29–41.

1.1 Learning communities

Grounding the educational and organisational model on the social constructivism and on the situated learning approaches, the new on-line communities intend to constitute themselves as experience centres, in which learning is not separated from action and where the learning process is more aimed at the community than at the individual. In this sense, Wilson & Ryder [2] refer that the learning communities are alternative metaphors to the traditional education systems. Thus, we consider that learning communities on the Web are alternative educational approaches to the traditional teaching systems, in which the perspectives of development of methods and universal strategies oriented to an effective, sequential and centralised teaching cannot capture the nature of the model of activity in the constructivism approaches.

The organisational model of the community and its functioning promotes the transfer of orientation, control methods and strategies of the learning development for its members, transforming it into a suitable complex system whose first manifestation takes place through multidimensional exploration of meaning construction and in the individual and collaborative adjustments towards a dynamic development of cognitive restructuring.

According to Wilson and Ryder, [2] this development process is oriented to real and authentic learning situations and includes: the self-organisation in functional communities with a common aim of learning support; the distribution of the orientation functions and of the control of the community activity by the members; and the emergence through the interactions among the group of the roles and member-specific functions.

The nature of the networked interactions on the Web constitutes a favourable environment for the development of learning communities, as the networked principle not only efficiently supports the plan of the interaction dynamics among its members, without limits of time or distance, but it facilitates the development of communicative processes with a collaborative nature in the course of the activities and experiences in knowledge building as well.

2. INTERACTION IN LEARNING COMMUNITIES

The absence of centralised control and the mutual influence among the community members follow a model of non-linear interaction, as a network of multiple representations and interpretations that substitutes itself to the logic of the singular and sequential representations in the traditional centralised educational environment.

This framing proposes a reading of the emergent complexity of the non-linear processes in communication and formation systems of the knowledge representations in learning communities, in particular in the absence of a centralised control and in the high interaction and mutual influence among the community members. It is through the interaction that we can observe an appreciation of the collaborative performance in the emergent modes of representation in the learning communities.

The educational development approach through the complexity theories, namely in the organisational activities of learning communities, tries to capture the complexity of the phenomena and non-linear processes in the mental activity, proposing a new platform for the conceptualisation and modelling the interactive learning and communication environments. This framing, as Tennyson & Nielsen [3] refer, is based on the conception that non-linearity constitutes a characteristic of the thinking and learning process.

The new education environments should allow the students to question their ideas and beliefs, stimulating in this way the development of an interactive and provocative process in the personal knowledge construction [4]. The result of this learning process shows that knowledge construction extends to a variety of sources, from the plan of the collaborative interaction among the community members, to the interaction plan between the student and the *knowledge media,* through the multidimensional exploration of knowledge representation.

According to what has been written, the communication interaction is developed beyond the sharing of knowledge representations, characteristic of the transmission model, to transform itself in a constructive process of meaning making, supporting the efforts and the involvement needs towards a social participation of the members of the learning community [5].

3. KNOWLEDGE MEDIA

The knowledge media, according to Eisenstadt [6], establish a new development model in the relationships between people and knowledge, because they are dynamic and they promote, through this characteristic, the possibility of a deep interaction with knowledge representations. The combination of computer science and telecommunications makes clear that knowledge media are more than means of contents presentation, transforming them in a fundamental condition for the learning communities formation [7].

The dynamics of the learning media is an emergent characteristic of the interaction facilities with the networked representations, whose process transcends the regulating logic of the information delivery of traditional media. We can observe this in the processes of transmission in the traditional paradigm of educational communication.

The knowledge media favours the individual's participation in the sharing of assumptions, beliefs, perceptions and complex representations. In this way, they describe the communication aim, the collaborative knowledge construction that develops when people communicate with each other, through and with the media. In other words, this is a means of transforming information into knowledge through the collaborative sharing of the individual representations and multiple perspectives.

The dynamic perspective of knowledge development in the individual is clearly underlined by the cognitive sciences, which describe it as a complex interactive network of information and competencies, which is developed in the knowledge media through the mutual implication between the media and the learning constructive process, acting like this as an expansion of the student's cognition and interaction networks.

The current cognition approaches are based on dynamic and adaptive phenomena. This approach includes the self-regulation capacity or learning monitoring, the development of the representations restructuring and perspective learning as a constructive process. From this point of view, we must acknowledge the fact that the cognitive system components (or the subsystems) support multiple processes of flexibility and adaptation, which find a favourable atmosphere to its development in an approach based on the network of the interactive media.

The knowledge media are the means to the development of flexible and adaptive representation models, whose importance is fundamental to the understanding of the functioning of learning communities on the Web.

4. FLEXIBILITY AND HYPERTEXT REPRESENTATION

The concept of flexibility starts from the non-linear or non-sequential organisation of information proposed to the hypertext systems, by Ted Nelson in the sixties. Based on this emergent theory whose application fields include a growing variety of domains, from communication to learning on the Web, also including the Web architecture, the representation is not a rigid structure, but, on the contrary, a dynamic body of information which shows itself under multiple shapes and suggests, at the same time, multiple exploration paths.

From this point of view, a piece of information is in the origin of a variety of texts, multiple narratives and interpretations, all of them coming from the network which supports the model of information representation and the new place of the reader (user). These (readers as users) move from the traditional perspective in order to be encouraged to intervene in the interacting representation model. In other words, when exploring the network, the hypertext user will be building their own hypertext, according to their previous knowledge, needs and aims.

The hypertext representation is deeply plastic, because the reader develops their own construction through the multiple courses explored by them having larger responsibility in this process. This derives from the exercise of adequate paths selection towards the construction of their own understanding model. This growing autonomy has implications in the students' learning activity and style. While for the traditional media it is expected that the author establishes a coherent and sequential organisation of the ideas and contents presentation that the readers should follow, in the hypertext we find a network of ideas and contents with multiple presentation paths, as well as multiple representation dimensions for the relationships among the ideas. According to this, the hypertext flexibility is the means for the flexibility of learning in the meaning making, as the hypertext network supports the dynamic process of the relationships formation.

According to Borsook [8], the traditional media discourage this activity, particularly at the exploratory level of the representational dimensions, leading the subject's intervention towards a role of passive acquisition of externally organised models and frequent simplification of the relationships complexity among the information nodes. In the sequence of this perspective, the tendency for simplification which stands out in the development of the learning facts and not in the formation of the relationships among these, remove the student's capacity of focusing their attention on the relevance of the relationships among the ideas. Supporting this type of activity, the hypertext not only presents the information, as the traditional media usually do, as the book, but it also represents it, since it becomes the medium for the development of the relationships network.

The hypertext network acts itself as a semantic network of knowledge representation formed by nodes and links, to which respectively correspond the concepts and the relationships among these. To the hypertext representation, the nodes include from small units to blocks of textual or graphic information and a connection structure that is processed through the entities close to the semantic network links.

The inter-connection among the information units performed by the student, implies that the representation built by them will be an emergence of the complexity of the relationships in the hypertext network, transforming the learning process, according to Jonassen [9], in a true expansion of the student's semantic network. The hypertext representation is, though, an expression of the dynamic pattern of the relationships established among the multidimensional nodes, and so a way of understanding complexity. In this sense, Landow [10] refers that the traditional texts are like limited languages, in which all the parts are known, but not the potential of their combinations, whose complexity is experienced by the reader in the hypertext network.

The authoring perspective in hypertext is based on the relationships construction activity. On the other hand, this dimension characterises the development model of the educational hypertext environments as it acts as the main tool for the construction of the individual and collaborative representations in the learning process and knowledge understanding, only possible through the profound interaction with the hypertext network. The interaction among the hypertext representation system and the student is a process of continuous participation and experimentation in knowledge building, transferring the learning control to the student, through the sustained exploration of knowledge understanding and of its development context.

Flexibility, the multidimensional representation and the learning contextualisation characterise the development dimensions of the hypertext environments that support virtual learning communities, namely in the implication of the experience and of the previous knowledge in the learning transformation process in an expansion of the student's representation models, and in the sharing of these models with the members that participate in the learning contexts. In this perspective, the hypertext environment constitutes an individual and collaborative learning experience zone.

5. SITUATED LEARNING AND HYPERTEXT ENVIRONMENTS

One of the most significant changes in the current learning theories proposes that knowledge should be observed not as an abstract and out of context representation, but as a constructive process that emerges from specific situations and contexts [11, 12, 13].

According to Clancey [12], the situated cognition theory is based on the fact that all the thoughts and human actions are adapted to the environment, being located. This way, what people understand, the way they conceive their activities, and what they do in physical terms, is developed in a conjunct construction.

In this sense, Wilson and Myers [14] refer that knowledge, learning and cognition are social constructions, expressed in actions of people who interact in the midst of the communities.

The participation is the main element for cognition and for the situated learning, because it requires the development of the negotiation in the process of meaning construction in the different situations and contexts in which it happens [13]. This process, according to these authors, implies that understanding and experience are in continual interaction, and that the participation notion decreases the distance among contemplation and involvement, abstraction and practice, being, this way, actions, people and the world implied in the thought, in the discourse, in knowledge and in learning, thus accomplishing an immersion process in the contexts of knowledge construction.

The environments that result from this new conception are defined by the learning contextualisation, by the individual decision on the materials to work with, by the identification of the aims to reach and by the community's involvement in the definition of a strategy for the construction and experiencing of the situations and contexts of knowledge production. This approach stands back from the learning conception based on the systematic knowledge acquisition and retention, and competencies externally defined that limit learning to the activity of internal information processing accomplished by the individual [15, 16].

The learning perspective based on the information processing metaphor focuses on the knowledge structure and on the structure of cognitive processes, to receive the information and to proceed to its integration in the existing structures, modifying them in order to accommodate the new information. However, and according to this view, the learning thus accomplished frequently results in knowledge isolated from the remaining representations in the mind. This type of knowledge is often referred as inert knowledge, generally not used out of the initial acquisition context [17, 18, 1]. In this sense, the symbolic cognition approach or information processing focus on the individual information processing activity, separated from culture and from the real learning contexts, treating information as a neutral construct.

However, it is through the development of the participation and collaborative exploration activities of real and authentic environments that the real life learning processes emerge [1].

The environments based on the theories and technologies of the hypertext and hypermedia representation flexibility constitute a support for the learning contextualisation. The exploration activity sustains the learning accomplishment through the relationship established by the individual and the community members with the different aspects of the social or physical situation, promoting the understanding of the multidimensionality of the representation and of the social factors involved. More than an integration of the individual in the environment, hypertext draws a deep interaction between the individual and the environment.

The development of strategies directed to the collaborative participation and exploration in significant contexts and authentic learning environments are the challenges of this approach.

In the hypertext network representation, the flexibility is the means for the conception of environments. Through these, it is possible to create and to simulate significant learning contexts, according to the principles of the situated learning, favouring the exploration of the several knowledge dimensions, promoting the observation of alternative points of view through the multidimensional exploration of the knowledge representations, confronting the knowledge building with authentic situations, and understanding the problems that the experts find in several areas and the knowledge that these experts use for solving them. In this sense, the hypertext environments are supports for the promotion of the development of cognitive flexibility in the acquisition, organisation and knowledge transfer to new situations and contexts.

The development of competencies of flexibility in the learning process and the creation of knowledge representation models that support cognitive flexibility require flexible learning environments that allow the presentation and the learning of the knowledge items in a non-linear, relational and multidimensional process, favouring the exploration activity, cognitive restructuring and knowledge transfer [19, 20, 21, 22].

The development of the understanding of knowledge representations results from the exploration of the multidimensionality of the information network, through the continuous exploration and construction of the meaning dimensions which took place in the collaborative environments on the Web.

6. COLLABORATIVE LEARNING ON THE WEB

The Web is the virtual place for the development of hypertext-based learning communities through which the interactions with the knowledge distributed in the network take place. Web based learning is deeply influenced by the virtual nature of the social interactions, and also by such factors as technology and the instructional practices of these learning communities.

The learning community on the Web implies the formation of a flexible and distributed knowledge representation environment, in which the hypertext approach and hypermedia technologies not only are one of the means for the information organisation and for the representations in the network, but also the means of development of extremely powerful collaborative environments for the accomplishment of the learning and for the active knowledge construction. The basic assumption of on-line collaborative learning is to build a learner-centred learning community, that treats the learner as an active participant [23].

From this point of view, the Web is a means to assist the learning process. In this process the students navigate in the multidimensionality of the flexible and distributed representations, establish relationships among the contents and the community members, through which they participate in a collaborative learning process. The hypertext nature of these environments promotes a wider control of the student on the learning experiences, and also points out the importance of the critical role of the individual aims in the quality and nature definition of the accomplished learning experiences [24]. This process promotes the reflection on the new knowledge under the form of the continuous negotiation of the individual representations, that displaces now to the sharing of thought patterns in the learning communities.

The sharing of ideas that emerges from this new dynamics of the networked communication, constitutes the central practice and the defining line of these new virtual learning, communities [7]. The challenges of this new approach are reflected necessarily in the nature and development of the virtual learning community activities, both in the plan of its definition as a group, and in the development of the relationship with the knowledge.

The notion of virtuality is fundamental for the understanding of the organising link in the new learning communities. The absence of the traditional notion of the community sense linked to the physical place is in this case replaced by the sharing in the knowledge construction and renewal among the community members, accomplished through the interactions in the network, not taking into account the physical dimension of the territory and the place as defining traits of the group identity. In this sense, Lévy [25] refers that this is a community that builds the social bond through the relationship with knowledge.

An effect of the organising link in virtual communities is the development of a culture of simulation, namely through the process of continued renewal of the social bond with knowledge in the Web communities and, this way, in the construction of a collective intelligence through the collaborative practices of the distributed learning. According to Lévy [26], the culture of simulation is a specific form of knowledge characteristic to the cyberculture that emerges from the networked practices of communication on the Web, drawing the contemporary process of mutation regarding the relation with knowledge, underlining the importance of sharing virtual worlds and complex universes of meaning.

As a support of the virtual groups that constitute the learning communities, the Web is also the medium for the development and proliferation of local narratives, individual stories and communicational fragments that interweave in an obscure network of authors and readers that constitutes the expression of the collaborative knowledge construction in a community of interests, aims and shared experiences.

The dynamics of the interaction processes which take place in the authors and readers network suggests a diversity of forms through which learning can happen in a community. Though, and according to Wilson and Ryder, [2], the learning community activity tends to produce an interaction pattern mutually sustained that defines the learning collaborative support in the Web communities. The following aspects, proposed by Sherry & Wilson [5], characterise this collaborative model: *i) articulating the learning need; ii) seeking help in a group forum; iii) engaging in a help consultation; iv) assessing learning; v) sharing the solution with the group; vi) archiving the interaction or restated solution for future reference; vii) repeating this process, or any part, if necessary to support learning.*

The collaborative learning on the Web stresses the mutual engagement, the sharing of information and multiple perspectives, and the joined knowledge building activity in the networked communication practices among the community members.

The creation of a favourable environment to the involvement of all the members in the community activities implies, in the participatory processes of search and help consultation, a form of immersion in the knowledge representations amongst its members. Knowledge sharing through the electronic mail, the audio and video conference, the discussion group and the forum promote the progressive involvement of the community members in the process of negotiation of knowledge representations, a continuous readjustment of the models, an understanding of the knowledge complexity and still the development of the critical thinking through the shared experience. Besides, the networked communication on the Web transforms itself in a expansion of the cognitive capacities of the students.

It is in this sense that McLellan [27] approaches the importance of the cooperative and sharing processes in the networked communication and learning, according to the collaborative model of Schrage. This model is based on the participatory principle that exists in the construction of the shared experience that describes a dynamic and active process, like the one that happens in face-to-face conversation, or in the on-line conference, being the *experience* the context of production and knowledge dynamic renewal. In opposition, the sharing of an experience or event acquires a passive nature closer to the information reception phenomena.

The sharing principle is fundamental to the formation of the networked interrelated ideas, strategies and theories, the ones that Romiszowski [28] refers are essential to the process of critical analysis, knowledge assessment and the new knowledge creative synthesis.

7. CONCLUSION

Learning in the Web learning communities is based on the networked interaction and communication and on the collaborative processes in the experience and knowledge building. The organisational bond of learning communities develops itself in the relationship with the knowledge through the collective and strongly interactive communication, among the individuals and between these and the distributed representation systems. The collaborative aspects are characterised by the mutual involvement in the community activities, in the knowledge sharing among its members and in the joint participation in the learning construction.

The community's members develop networks of flexible and collaborative knowledge construction, based on an active learner model, led to the promotion of the individual initiative and autonomy in the exploration of the representational multidimensionality of the distributed knowledge. So, it is an activity model that it is drawn from the individual to the cooperative, through the mutual engagement of the community members in the process of the knowledge building and the sharing of multiple perspectives of thinking. The activity of the learning community is a support for the amplification of individual cognitive capacities, as those of the group, in the development of the creative processes, in the critical thinking and in the collaborative work.

REFERENCES:

1. Rogers, J. (2000). Communities of Practice: A framework for fostering coherence in virtual learning communities. *Educational Technology& Society.* 3 (3), 384-392.
2. Wilson, B. & Ryder, M. (1998). *Distributed Learning Communities: an Alternative to Designed Instructional Systems.* http://www.cudenver.edu/~bwilson/dlc.html (24.6.2000).
3. Tennyson, R. D. & Nielsen, M. (1998). Complexity Theory: Inclusion of the Affective Domain in an Interactive Learning Model for Instructional Design. *Educational Technology*, XXXVIII (6), 7-12.
4. Parker, A. (1999). Interaction in Distance Education: The Critical Conversation. *Educational Technology Review*, 12, 13-17.
5. Sherry, L. & Wilson, B. (1997). Transformative Communication as a Stimulus to Web Innovations. In B. H. KHAN (Ed.), *Web-Based Instruction.* Englewood Cliffs, N.J.: educational Technology Publications.
6. Eisenstad, M. (1995). *Overt Strategyfor Global Learning.* London: Kogan Page.

7. Berg, G. A. (1999). Community in Distance Learning Through Virtual Teams. *Educational Technology Review*, 12, 23-29.
8. Borsook, T.K. (1997). Hypermedia: Harbinger of a New Instructional Paradigm? In C.R. DILLS & A.J. ROMISZOWSKI (Eds.), *Instructional Development Paradigms*. Englewood Cliffs, NJ: Educational Technology Publications.
9. Jonassen, D.H. (1990). Problems and Issues in Designing Hypertext / Hypermedia for Learning. In D. H. JONASSEN & H. NANDL (Eds.), *Designing Hypermediafor Learning*. Berlin:Springer-Verlag.
10.Landaw, G. P. (1994). What's a Critic to Do?: Critical Theory in the Age of Hypertext. In G.P. Landaw (ed.), *Hyper/Text/Theory*. Baltimore, Maryland: The Johns Hopkins University Press.
11.Brown, J.S., Collins, A. & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.
12.Clancey, W. J. (1997). *Situated Cognition, On Human Knowledge and Computer Representations*. Cambridge: Cambridge University Press.
13. Lave, J. & Wenger, E. (1991). *Situated Learning, Legitimate Peripheral Participation*. USA: Cambridge University Press.
14. Wilson, B. & Myers, K.M. (1999). *Situated Cognition in Theoretical and Practical Context*. http://ceo.cudenver.edu/~brent_wilson/SitCog.html(14.5.2000).
15.Hannafin, M.J., Hall, C., Land, S. & Hill, J. (1994). Learning in Open-Ended Environments: assumptions, methods, and implications. *Educational Technology*, XXXIV (8), 48-55.
16.Orey, M.A. & Wayne, A. N. (1997). The Impact of Situated Cognition: Instructional Design Paradigms in Transition. In C. R. DILLS & J. ROMISZOWSKI (Eds.), *Instructional Development Paradigms*. Englewood Cliffs, N.J.: Educational Technology Publications.
17.Bransford, J., Cunningham, D., Duffy, T.M. & Perry, J.D. (1990). Anchored instruction: Why we need it and how technology can help. In D. NIX & R. SPIRO (Eds.), *Cognition, Education, and Multimedia*, Hillsdale, NJ: Lawrence Erlbaum Associates.
18.Resnick, L. (1987). Learning In School arid Out. *Educational Research*, 12(9), 13-20.
19.Dias, P. (2000). *Estilos e estratégias na interne, Web: dimensões de desenvolvimento das comunidades de aprendizagem*. Comunicação apresentada no seminário CENTED 2000/Viagens Virtuais, Universidade Aberta, Lisboa, 10-12 de Janeiro.
20.Dias, P., Gomes, M.J. & Correia, A. P. S. (1999) Disorientation in Hypermedia Environments: mechanisms to support navigation. *Journal of Educational Computing Research*, 20 (2), 93-117.
21.Jacobson, M. J., MAOURI, C., MISHRA,'P. & KOLAR, C. (1996). Learning With Hypertext Learning Environments: Theory, Design, and Research. *Journal of Educational Multimedia and Hypermedia*, 5 (3/4), 239-282.
22.Spiro, R., Feltovich, P., Jacobson, J. & Coulson, R. (1995). Cognitive Flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Acquisition in Ill-Structured Domains. In L.P. STEFFE & J. GALE (Eds.), *Constructivism in Education*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
23.Harasim, L., Calvert, T. & Groeneber, C. (1997). Virtual-U: a Web-Based System to Support Collaborative Learning. In B. H. KHAN (Ed.) *Web-Based Instruction*. Englewood Cliffs,N.J.Educational Technology Publications.
24. Wilson, B. & Lowry, M. (2000). *Constructivist Learning on the Web*. http://ceo.cudenver.edu/~brent_wilson/WebLearnin.html (13.10.2000).

25.Lévy, P. (1994). *A Inteligência Colectiva. Para uma Antropologia do Ciberespaço.* Lisboa: Instituto Piaget.
26.Lévy, P. (1997). *Cyberculture, Rapport au Conseil de l'Europe.* Paris; Éditions Odile Jacob/Éditions du Conseil de l'Europe.
27.Mclellan, H. (1997). Creating Virtual Learning Communities Via the Web. In B.H.KAHN (Ed.), *Web-Based Instruction.* Englewood Cliffs, N.J.: Educational Technology Publications.
28.Romiszowski, A.J. (1997). Web-Based Distance Learning and Teaching: Revolutionary Invention or Reaction to Necessity? In B.H. KHAN (Ed.), *Web-Based Instruction.* Englewood Cliffs, N.J.:Educational Technology Publications.

PAPERS

PART II

Evaluation Criteria for Hypermedia Educational Systems

Ignacio Aedo, Paloma Díaz
Laboratorio DEI. Departamento de Informática
Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911 Leganés (Spain)
{aedo@ia, pdp@inf}.uc3m.es

Key words: hypermedia, education, usability, evaluation criteria.

Abstract: Educational systems, as highly interactive applications, need to be assessed to determine if they are useful and usable. Evaluating an educational system involves testing its user interface as well as studying the underlying didactic method, the instructional materials and the interaction mechanisms to prove if they help users to reach their goals. With this purpose, educational developers need to rely on a set of clear, concrete and measurable quality criteria. In this paper, we propose a number of evaluation criteria for educational hypermedia systems that are based on the authors' experience as developers of educational systems, such as CESAR, Now-Graduado or CIPP. Such criteria are aimed not only at helping educational developers to evaluate their systems but also to provide them with some guidance to address educational requirements during the design process.

1. INTRODUCTION

Evaluation is an essential activity in the development process of any interactive system inasmuch as it provides valuable and reliable information about the product utility and helps to detect deficiencies and improvements. In fact, authors like Preece et al. [1] suggest that evaluation plays such a central role in the development cycle that it has to be applied in all its stages to understand and take into account the users' needs. This approach, usually referred to as user-centred design [2], requires involving the user in the development process and perform empirical tests that corroborate if the design decisions are appropriate.

Educational systems, as highly interactive applications, need to be assessed to determine if they are usable and useful. Such evaluation is not only concerned with assessing the quality of the user interface that can be tested in terms of such parameters as readability, aesthetics or consistency; the evaluation process is also oriented towards analysing if the system can be used in an efficient way to meet the needs of its users [3], who can be students and teachers. To perform such evaluation various methods can be applied, including analytic, expert, empirical and experimental procedures [4]. The decision of using either one method or another mainly depends on such factors as the development stage reached or the available resources. Thus, analytic methods, which make use of formal or semi-formal descriptions of the system, are essentially applied during the analysis or early design stages whereas experimental methods require a fully developed prototype. In any case and before selecting a specific method, developers have to establish a set of clear, concrete and measurable criteria to assess the system [5].

Taking as starting point research works on hypermedia evaluation criteria [6, 7, 8, 9, 10, 11] and our experience in the development of rather different educational hypermedia systems, such as CESAR, Now-Graduado and CIPP, in this paper we are studying how to extend and apply evaluation criteria to analyse and improve the usability and usefulness of instructional systems. The set of criteria proposed here is aimed at making up a useful framework for evaluating and comparing educational systems and at providing some guidance to address educational requirements during the design process.

2. RELATED WORKS

Evaluation criteria provide a useful framework to assess the quality of a system in terms of utility, usability and maintenance [11], to compare similar applications or to detect problems and improvements. Indeed, the development of a set of evaluation criteria for hypermedia systems has attracted the attention of several researchers.

Thus [6] proposed the use of two levels of criteria, global and node, that provide authors with some feedback about the structure of their hypertext with a view to detecting possible navigation problems. The first group includes other two criteria, compactness and stratum, that study properties concerning the whole hyperdocument while the second group comprises two criteria measured at the node level: depth and imbalance. In both cases, only structural and mathematical properties of the hyperdocument and its components are taken into account to try and detect errors, such as unreachable nodes, which decrease the utility of the navigational structure. Two structural parameters are also proposed by Hatzimanikatis et al. [8]: readability and structure maintenance. In this case, the hyperdocument is studied in terms of size, path complexity (i.e. number of different paths or cycles which can be found in a hypertext graph), tree impurity (i.e. the extent to which a graph deviates from being a tree), modularity (i.e. whether modules are self-contained and independent), individual node complexity (i.e. the complexity that a single node imposes on the overall hypertext structure), coherence, complexity and simplicity of the contents. However, and as pointed out by Yamada et al. [9], those criteria do not provide any kind of hint about the navigation structure utility. In this line, Yamada et al. [9] present three metrics oriented towards analysing how users feel like when navigating through hypertext: interface shallowness, downward compactness and navigability. Again this set of well-defined criteria is focused on analysing structural bugs which can scale up the 'lost in hyperspace' problem.

In any case, the aforementioned approaches can only be applied to evaluate hyperdocuments in order to improve the navigation through the hyperdocument, but apart from structure, hypermedia applications have other dimensions that have to be tested during the development cycle, including the contents, their presentation features or the interaction mechanisms. Moreover, such approaches are intended for providing hypertext authors with objective data that they have to analyse to decide if the hyperdocument satisfies their expectations. However, hypermedia authors also need a set of quite different parameters capable of gathering information about the system utility from the user's point of view. In contrast to measures like node depth or reachability, usability aspects cannot always be formally and mathematically expressed since they do not strictly depend on the hyperdocument elements or relationships but on the users' opinions about the final product. In fact, developers can follow some recommendations to increase the quality of their applications but at the end they have to go through an empirical evaluation to prove if the system fulfils the users expectations and needs.

Evaluation criteria to assess the hyperdocument usability were introduced by Garzotto et al. [7], including richness, ease, consistency, self-evidence, predictability, readability and reuse, and they were extended by Ficarra, [10] to offer hypermedia developers a useful framework to perform their evaluations. As it can be seen, a mathematical rule cannot be defined to establish how easy to use a hyperdocument is according to the number of nodes and links it has. Such kind of criterion has to be empirically tested to have some evidence about its accomplishment. Moreover, apart from these usability criteria other aspects affecting the hyperdocument acceptance can be considered. Indeed, such factors as performance effectiveness, error tolerance or system integrity affect the system usability [3].

Some work on criteria for hypermedia educational systems was done by Mendes et al. [11], although the main aim of this proposal was to assess the system in terms of the reusability of information, maintainability of the application and the development effort. Even though these parameters make it possible to analyse some relevant features of any kind of software application there is not a direct relationship among them and the educational utility of a system. In fact, a system can be completely structured, modular, easy to maintain and have no educational value.

In any case, all these criteria made up a useful framework to evaluate and compare systems but some developers, and particularly those who are novices, also need concrete guidelines to improve their systems taking into account these criteria during the development cycle. In this context, we propose a number of evaluation criteria for educational hypermedia systems that are based both on the works described in this section and on the authors' experience as developers of educational systems, such as CESAR, Now-Graduado or CIPP, all of which have rather different educational goals and target users. Thus, while CESAR [12] is a hypermedia learning environment based on electronic stories that is aimed at helping hearing-impaired children learn the signed and written languages, Now-Graduado [13] covers the most difficult subjects at primary schools and it is intended for low-qualified women who are not used to computers and have forsaken their study discipline for years; and, finally, CIPP [14] is a hypermedia electronic book that was developed to assist Computer Science students in learning the Pascal programming language.

3. EVALUATION CRITERIA FOR EDUCATIONAL HYPERMEDIA SYSTEMS

In this section we propose a number of evaluation criteria for educational hypermedia systems which can help educational developers in two ways:

- evaluation criteria make up the basis to perform a formative evaluation, which is mainly oriented towards exploring if the system is satisfactory or needs some changes [15].
- they also provide some kind of guidance to software developers when trying to address educational requirements (e.g. how to increase the didactic material richness).

To evaluate an educational hypermedia system developers need a number of clear, concrete and measurable criteria to determine whether the system is useful and usable at the same time. On the one hand, they have to assess the interface quality to detect usability problems or misunderstandings that can be solved to improve the interaction process. But the evaluation process has also to focus on the system utility, so that it has to be analysed whether it can be used in an efficient way to meet the needs of its users [3]. It is worth noting that users of educational systems can be students and teachers, both of whom pose rather different requirements [16]. While students need to acquire a procedural knowledge and to develop practical skills interacting with the courseware, teachers will be willing to imprint a specific didactic method. Thus, evaluating an educational system involves assessing the user interface usability as well as the utility of the underlying didactic method, the instructional materials and the interaction mechanisms to prove if they help users to reach their learning and teaching goals. In the next subsection criteria to assess both the user interface and the educational usefulness are described thoroughly.

3.1 Evaluating the usability of the user interface

The user interface, as the communication channel through which the user comes into contact with the computer, has to make it possible to perform the users' tasks. A great bulk of recommendations has been proposed to produce more usable user interfaces that take into account the physical and mental capabilities and limitations of users, such as in Smith [17], but the only way to prove that the design decisions are right is to empirically evaluate its interface. In this subsection we describe a number of criteria to evaluate the usability of educational hypermedia systems, including aesthetics, consistency, self-evidence, naturalness of metaphors and predictability.

3.1.1 Aesthetics

It studies how the inclusion of multimedia information is coordinated and used to enhance the legibility and comprehension of concepts. Aesthetics is intended for analysing if multimedia contents are well organised in the visualisation space and synchronised and if they are always understandable, that is, they can be always perceived by any user. In this way, this criterion comprises aspects such as motivation [10] or readability and it is extended with the concept of appropriateness. Educational developers should take into account if the presentation features of their applications not only make the hyperdocument understandable but they are also pertinent for the users. For instance, in Now-Graduado the appropriateness of the aesthetics was considered as a crucial factor to get a positive attitude from users. Even though the subjects treated in this system were quite simple, since they cover primary school concepts taught to children between 5 and 14 years old, Now-Graduado users were adults and, hence, graphical designers had to take special care over the design of icons, images and the rest of multimedia contents to avoid an infantile appearance that would make users felt undervalued.

Some aspects that have to be taken into account to analyse the system aesthetics are:

– The global look and feel of the application: it must be appropriate for the target users taking into account their special characteristics (e.g. age, social and cultural background, disabilities) and needs.
– The legibility of the contents: all the items included in each node must be presented taking into account the capabilities and limitations of users [18]. That affects such features as the size, style and colour of texts, the length of text lines and the spacing, the quality of images or the pitch and loudness of sounds.
– The rhythm of presentation of multimedia contents: contents with an implicit duration (e.g. video or sound) must be synchronised with the users' pace of assimilation.
– The density of each node, that is, the quantity of items it includes: developers must include enough information into each node without saturating the users' capability of assimilating information.
– The contrast of the colours used in the interface: positive contrasts (dark foreground- light background) are more readable than negative contrasts (light foreground - dark background) [19].

3.1.2 Consistency

It determines to which extent elements that are conceptually similar are equally treated by the application while those that are different are differently treated [7]. Consistent educational applications are easier to use and remember and, therefore, users can pay more attention to reach their learning or teaching goals than to learn how to use the system. Interface consistency chiefly consists of using the same appearance for the links ant the other facilities, so that the user can recognise them easily. If the hyperdocument's author decides to use a specific visual clue (e.g. colour, font, style) to identify the links of a node as well as their status (e.g. visited or not), he/she should not use this resource with other purpose such as highlighting text since it could provoke unnecessary user errors.

Some aspects that have to be taken into account to analyse the system consistency are:

- Interface areas: the node visualisation area can be organised into several areas, each of them has to be used always for the same purpose. For instance, areas like functions/buttons, contents and messages can be identified in many interactive applications. If the message area is always used to show information about the system status it should not also provide other kinds of information such as location clues. CESAR, Now-Graduado and CIPP were designed using interface areas so that the users know in advance what they can expect in each area of the window and where they have to look for each kind of information.
- Labels and messages: the communication in written or in spoken form with the user has to be consistent so that in the same situation the same label or message has to be used.
- Buttons and icons: both elements are often ordered following some criteria, whether alphabetically or trying to minimise their semantic distance, and they are also arranged in a unique bar. The button/icon bar size and appearance have to remain stable, so that the user always see the same set of functions placed in the same positions. If an action is disabled it is more consistent to show the corresponding button/icon using a visual attribute to indicate its status than to dynamically change the bar aspect.
- Visual clues: whenever an attribute of an object is used as a visual clue (e.g., colour, background) it has to be applied in a consistent way in all the nodes where the object appears. For example, underlining text in web-based applications can be considered as a consistency error since underlined words are supposed to be links.

3.1.3 Self-evidence

Self-evidence determines how easily users can guess the meaning and purpose of everything they are presented [7], and it mainly analyses how tangible the structure of the system and its functions are. For example, in an educational system it can be evaluated if the user knows how to interact to solve a particular exercise and he/she understands the purpose of such an exercise. This criterion also comprises the concept of transparency of meaning [10] which analyses how the concepts concerning a particular content are not ambiguous.

Some resources that can be used to increase self-evidence are:

- Metaphors: interface metaphors have been used in many educational systems to try to increase the hyperdocument tangibility. The main principle underlying interface metaphors is to translate a user's model of a process into the computer model which implements such process, so that users are supposed to be familiar with each object and function.

However, as it is discussed in next subsection, the use of metaphors has provoked some controversy in the HCI community.
- Self-contained nodes: that users can identify easily.
- Expressiveness of multimedia items: designers can profit from multimedia to make the nodes easier to recognise.
- Coherent and useful links: an excessive number of unnecessary links hinder the understanding of the hypertext structure.
- Meaningful naming of functions: the actions users can perform should be identified by means of a name that belongs to the application domain. Moreover, the terminology used in texts as well as in the system commands and dialogues has to be clear and unambiguous, so that the user can understand each message and concept. For example, to start a learning session a student can be asked to "Select a course" and "Study a lesson" instead of "Select directory" and "Open file".

3.1.4 Naturalness of metaphors

Interface metaphors, like the desktop, electronic book, story or the museum have been used quite often to try to approximate the user's model to the computer model. However, some authors do not agree with the supposed helpfulness of this resource [20, 21] which has to be adopted as far as it does not constrain the designers or the users. If the metaphor does not gather all the features of the domain or treat them in a different way it can constrain and mislead users [20]. Indeed, the naturalness of metaphors [10] can be measured in terms of the appropriateness of the use of metaphors to improve the communication with the user. For example, one of the most relevant conclusions of CESAR's evaluation was that the use of the book and the story metaphors was a good choice not only for usability purposes but they could also help to socialise children (e.g. learning to share books or providing them with the opportunity of being told stories).

Some aspects to be considered at this point are:
- The interface has to gather all the features of the domain of application which are required to reach the users' goals. If such features are not supported by the metaphor, it has to be increased.
- If the use of a metaphor to represent some features of the domain of application decreases the system efficiency or does not contribute to make the system more usable, then such metaphor has to be relaxed.

3.1.5 Predictability

This criterion represents to which extent users can anticipate a system outcome [7], that is, if users know which kind of result they will get from a specific interaction. Predictability is different from self-evidence, since users can identify the purpose and function of each object they are presented but they cannot be sure about what will happen if they perform an action. For example, users can be able to initiate a search (self-evidence) but they can ignore if they are going to directly move onto the first node including the search pattern or if they will be shown a list of results (predictability). The best way to increase the system predictability is to perform a task analysis with the users, to understand what they can expect from each interaction and how results are supposed to be presented.

Then, predictability can be increased taking into account the following aspects:

– Designers have to perform a detailed task analysis before designing the system involving the users in such study, so that they can define predictable interaction mechanisms.
– Between two presentation choices, designers must always select the one which is more meaningful in the domain of the application.

3.2 Evaluating the educational usefulness

Apart from the quality of the interface, educational developers have to test if their systems make it possible to reach the learning and teaching goals of their users. In this subsection we describe a number of criteria oriented towards assessing the educational usefulness of educational hypermedia systems, including richness, completeness, motivation, hypertext structure, autonomy, competence, authoring, adaptation, flexibility, communication support, collaboration support and accessibility.

3.2.1 Richness

This criterion expresses the volume of information included in the system as well as the different ways offered to reach it [7]. For instance, richness is analysed answering sentences like "how many items are used to represent a particular concept?", "are all these items complementary?" or "which mechanisms are provided to the user in order to find information about this concept?". That implies that not only the contents comprised in the system have to be assessed but also the different options provided to the user to access them, whether using associative or structural links or navigation tools like search engines, maps, indexes and so on. For example, CIPP contains the set of syntactical diagrams of the programming language that can be consulted using three techniques: contextual navigation, which is done through the diagrams included in the theoretical explanations; structural browsing, which is based on the selection of links associated to the structure of a Pascal program; and direct access using the alphabetical index of diagrams. In this way, each student can choose the most appropriate access method depending on his/her learning goals.

This concept of richness can be extended to take also into account the different ways information is presented and used, a key feature in educational software applications devoted to meet the needs of different kinds of students. In fact, the same knowledge can be approached from quite different ways depending on the user's background, previous knowledge and other factors. Consequently, the richness of presentation styles of the application can also be measured to analyse its utility. For example, in CESAR students are told stories using different media (video/image supporting sign language, images or text) depending on their learning competence [22].

Moreover, the interaction richness is also a quite relevant parameter to be considered, in particular, for the exercises proposed to the students where they are incited to play an active role. Thus, different solving strategies can be proposed to different students to try and benefit from the student's learning style as in CESAR [16]. In this system, each exercise is divided in two parts: information objects and strategy. An information object is a composite entity that embodies different representations (called attributes) of a concept. For example, the concept of "house" can be defined using four attributes: the written form, the sign, an image and a definition. The strategy defines which attributes of each object are used, how the exercise statement is presented, how the student can interact to solve it and what to do when the student gives an answer. For instance, an exercise using the concept of house can consist of showing students the image of a house and ask them to select among a number of signs which one corresponds to the image. In a different exercise, students can be asked to write the word corresponding the sign form of a house. As it can been seen in these examples, the same object is used with different learning goals and, thus, exercises are adapted to the student's needs by means of the adequate solving strategy.

To sum up, to analyse the richness of their educational hypermedia systems developers have to take into account:

- The number of nodes and the number of information items they include.
- The different navigation paths and tools offered to the users.
- The diversity of presentation and interaction styles supported.
- The different kinds of exercises and interactive activities provided.
- The educational objectives that can be fulfilled with the system.

3.2.2 Completeness

This criterion defines if the system has enough contents and interaction mechanisms to reach the educational goals of different kinds of users. Completeness is far different from richness. While the later only relates to the number of elements included in the system, completeness is concerned with their appropriateness.

To be considered as complete, an educational software application should provide tools for students and teachers, both of which have quite different objectives when using the system. For example, educational systems have to support different kinds of learning activities including tasks such as reading, creative writing, problems resolution and self-evaluation. It is important to design the environment combining complementary activities, such as active-passive, creative-reactive and directed-explanatory [23]. This was a basic principle in the design of CESAR and Now-Graduado, where students are proposed several kinds of exercises that are solved using different strategies.

Moreover, some communication support among students and teachers can be required, particularly if the system is being implemented as a web application where users can profit from the benefits of a distributed and multiuser environment. For example, the first prototype of CIPP, called Hyper-Apuntes, was designed as a web application where students could contact teachers and had a board where they could communicate with other students. In this line, cooperation can also be a useful didactic resource depending on the subject to be taught and the characteristics of the users.

Some aspects to test when analysing the system completeness are:

– Learning activities: designers have to provide learning tasks to fulfil the learning objectives of students.
– Authoring support: it has to be considered to which extent the system can be modified by the users. On the one hand, students can enlarge the system editing, annotating and restructuring the didactic material or marking useful sections. On the other, teachers can require mechanisms to update the courseware or to adapt it to their students' needs.
– Communication support: if the system will provide communication mechanisms among users, some features that can be analysed are the number of communication tools offered (e.g., e-mail, chats, videoconferencing) and the level of communication provided to the users (e.g. synchronous vs. asynchronous).
– Collaboration support: collaboration is not only about communication and, therefore other services have to be provided including definition and management of working groups as well as different user roles, each one with different responsibilities and abilities to access information. Moreover, some technical requirements have to be met such as concurrency control, notification of events and safeguard of confidentiality and integrity.

3.2.3 Motivation

Another important aspect to assess in educational systems is the way in which students are motivated not only to use the system but also to learn more about the subject being treated. Thus, students need self-evaluation

activities reinforced with some kind of feedback that could help them to realise their errors. For instance, Now-Graduado includes not only several kinds of multimedia exercises but also at the end of each lesson students are invited to perform no computer-related tasks, such as visiting a museum or looking for a term in an encyclopaedia, that strengthen the learning process and encourage them to continue their learning experience once they have finished a computer session.

Adaptation can also be used as a resource to increase the student's motivation. Adaptive mechanisms or personalisation tools can be provided to make the system interface and function match the learning style of students [24]. For example, CESAR teachers can personalise the solving strategy of each exercise to each child, so that the one which better fits the features of that students can be selected.

To sum up, some aspects to take into account to increase the system capability to motivate students are:

- Providing self-evaluation mechanisms.
- Carefully designing the exercises feedback.
- Trying to relate the computer-based learning with other activities of the student's environment.
- Using adaptation of contents, interface or functions to adapt the software tool to the learning style of students.

3.2.4 Hypertext structure

This criterion is oriented towards analysing the structural properties of hypertext using parameters as those proposed by [6, 8, 9]. In this case, educational developers can obtain a number of objective measures that can help them to detect structural bugs. Parameters like the node reachability or modularity should be fulfilled by any hyperdocument for obvious reasons but others like the depth, imbalance, tree impurity or sequentiallity have to be considered in a more relaxed way since there is no empirical evidence on the need to satisfy them to produce a usable application. A sequential navigation based on the course hierarchical structure can be useful for novel students with no previous knowledge on the course subject as it was demonstrated in the evaluation of CIPP [14], but some educational developers will not agree with the idea of being compelled to generate a tree structure which can reduce the educational value of hypertext links. In the same way, those who conceive the learning process as a free exploration activity will not need to analyse whether the hypertext has a linear reading or not, irrespective of the fact that this aspect can make the navigation through hyperspace easier and that some studies reveal that free navigation does not imply a good learning process [25].

In this case, some aspects which can be considered are:

The hypertext has to be fully connected so that there is a hub node from which there is a path to reach all the nodes of the hyperdocument and vice versa. The existence of isolated nodes, which cannot be reached selecting links, reveals a negligent design.

- Modular hypertexts are easier to use and to maintain.
- Tree structures make navigation easier and help to make the hypertext structure explicit.
- A high degree of imbalance suggests that some subjects are treated less deeply than others.

3.2.5 Autonomy

Autonomy, which was defined by [10] as the degree of freedom of navigation offered to the user, can be extended to take into account the freedom of interaction which also includes the navigation. Concerning the freedom of navigation, there is a lot of controversy about the autonomy that has to be given to the students, but in general it depends on several factors including the students background and involvement. For instance, during the evaluation of CIPP [14], it was shown that most students made use of sequential reading since they thought it was a more efficient way to approach each subject for the first time than associative navigation. In any case, in order to avoid students getting lost in hyperspace, autonomy has to be complemented with other features (e.g. consistency, modularity, complexity of the node contents, etc.) and interface clues have to be provided to help students to orientate themselves.

Some aspects to consider are:

- The system offers an appropriate level of autonomy to navigate and interact. For example, Now-Graduado includes circular links that connect distinct views of a same concept. When selected, circular links open the target node disabling all its links. Users can read the node and interact with it, but at the end they come back to the source node.
- The system provides clues and helps to solve disorientation problems. For example, in CESAR, evaluators considered that associative navigation was not a problem since children could always count on a reference: the book.

– Different levels of autonomy are considered according to different kinds of users. For instance, associative navigation can be complemented with sequential access, as in CIPP.

3.2.6 Competence

Competence is related to the ability to navigate through the system and use it to reach a particular goal [10]. This parameter has to be carefully analysed according to the user's characteristics. For instance, an educational software application should have different levels of use to meet different degrees of expertise. Thus, novice users require clues and assistance mechanisms that tend to bother expert users. In any case, interactive help mechanisms are required, and such help has to be contextual so that users can receive the information they exactly need at each moment.

Moreover, it can be analysed how the system adapts to the different interaction styles or if the contents are appropriate for each type of user.

Some aspects to consider concerning competence are:

1. Different levels of expertise are supported.
2. Online contextual help is provided.
3. The system adapts to the user characteristics.

3.2.7 Flexibility

Flexibility is related to the facility to use and maintain the system.

First, the difficulties users will find to access the system have to be analysed including technical aspects (software and hardware platform required) as well as physical or personal features (e.g. disabilities, timing constraints, etc.). Then educational developers have to study the resources of their users to establish when and where the system will be used. Thus, web-based educational tools are highly flexible since they are supposed to be platform-independent and there are no time or distance constraints.

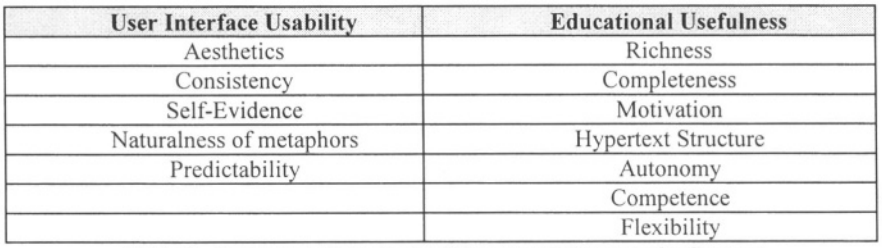
Secondly, maintenance has to be also taken into account and, therefore, the system has to be structured in a modular and quite structured way so that future improvements will be easy and cheap to afford. Moreover, the system architecture can be defined in such a way that modules can be reused for different applications so that the effort required to develop a new course can be considerably reduced.

To analyse the system flexibility designers can assess:

– Limitations and needs of users against the operation requirements of the system.
– Modularity and structure of the system architecture.

4. CONCLUSIONS

In this paper we have described the evaluation criteria for hypermedia educational systems which can be used to test the user interface as well as the educational usefulness of applications as summarised in Table 1.



User Interface Usability	Educational Usefulness
Aesthetics	Richness
Consistency	Completeness
Self-Evidence	Motivation
Naturalness of metaphors	Hypertext Structure
Predictability	Autonomy
	Competence
	Flexibility

Table 1: Evaluation Criteria for Educational Hypermedia Systems

REFERENCES

1. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T.. Human-Computer Interaction. The Open University. Addison Wesley, 1994.
2. Rubin, J. "Handbook of usability testing". John Wiley & Sons. Inc. New York, 1994.
3. Lee, S. H. Usability testing for Developing Effective Interactive Multimedia Software: Concepts, Dimensions and Procedures. Educational Technology & Society, vol. 2, n. 2, 1999.
4. Benyon, D., Davies, G., Keller, L. and Rogers, Y. A Guide to Usability- Usability Now!. Milton Keynes: The Open University, 1990.
5. Catenazzi, N., Aedo, I., Díaz, P. and Sommaruga,L. The evaluation of electronic books: Guidelines from two practical experiences. Journal of Educational Multimedia and Hypermedia. 6(1), pp. 91-114, 1997.
6. Botafogo, R.A., Rivlin, E. and Shneiderman, B. Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. ACM Trans. on Information Systems, vol. 10, n. 2, April, pp. 142-180, 1992.
7. Garzotto, F., Mainetti, L. and Paolini, P. Hypermedia Design, Analysis and Evaluation Issues. Comm. of the ACM, vol. 38, n. 8 (August), pp. 74-86, 1995.
8. Hatzimanikatis, A. E., Tsalidis, C.T. and Christodoulakis, D. Mesasuring the readability and maintainability of hyperdocments. Journal of Software Maintenance, Research and Practice, vol. 7. pp 77-90, 1995.
9. Yamada, S., Hong, J. and Sugita, S. Development and Evaluation of Hypermedia for Museum Education: Validation of Metrics. ACM Transactions on Computer-Human Interaction, vol. 2, n. 4 (December), pp. 284-307, 1995.
10. Ficarra, F.V.C. Evaluation of multimedia components. Proceedings of the International Conference on Multimedia Computing and Systems, Ottawa pp. 557-564, 1997.
11. Mendes, M.E.X., Harrison, R. and Hall, W. Applying Metrics to the Evaluation of Educational Hypermedia Applications. Journal of Universal Computer Science, April, 1998.

12. Aedo, I., Catenazzi, N. and Díaz, P., The evaluation of a Hypermedia Learning Environment: The CESAR experience. Journal of Educational Multimedia and Hypermedia, 5(1), 49-72. 1996.
13. Aedo, I., Díaz, P., Panetsos, F., Carmona, M., Ortega S. Huete E., "A hypermedia Tool for Teaching Primary School Concepts to Adults". IFIP WG 3.3 Working Conference "Human Computer Interaction and Educational Tools". Sozopol (Bulgaria). May 27-28, 1997,pp. 180-188, 1997.
14. Aedo, I., Díaz, P., Fernández, C., Muñoz, G. and Berlanga, A., Assessing the utility of an interactive electronic book for learning the Pascal programming language. IEEE Transactions on Education (forthcoming).
15.Cronbach, L.J., Course Improvement through evaluation. Teachers College Record, 64(8), pp. 672-683, 1963.
16. Díaz P., Aedo I., Torra N., Miranda P. and Martín M., Meeting the needs of teachers and students within the CESAR Training System. British Journal of Educational Technology. 29 (1). pp. 35-46, 1998.
17. Smith, S.L. and Mosier, J.N. Guidelines for Designing user interface software. Technical Report ESD-TR-86-278, The Mitre Corporation, Bedford, Massachusetts, 1986.
18. Faulkner, C. The essence of Human-Computer Interaction. Prentice Hall, 1998.
19.Nielsen, J. Designing Web Usability. New Riders, 2000.
20.Gentner, D. and Nielsen, J. The Anti-Mac interface. Comm. of the ACM, vol. 39, n. 8, pp. 70-82, 1996.
21.Hudson, W. Metaphor: a Double-Edged Sword. Interactions, vol. VII, n. 3, pp. 11-15, 2000.
22. Aedo, I., Martín, M., Miranda, P., Panetsos, F. And Torra, N. A teaching methodology for hearing impaired using hypermedia and computer animation. Journal of Computing in Childhood Education, 5 (3/4), 353-369, 1994.
23. Allinson, L. and Hammond, N. Learning Support Environments: Rationale and Evaluation. Computers & Education.15 (1-3), pp. 137-143, 1990.
24..Brusilovsky, P., Eklund, J. and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems. Proceedings of Seventh International World Wide Web Conference, 30 (1-7), pp. 291-300, 14-18 April 1998.
25.Sciarone, A.G. and Meijer, P.J. How free should students be? A Case from call: computer-assisted language learning. Computers & Education. 21 (1-2), pp. 95-101, 1993.

Creating Collaborative Environments for Web-based Training Scenarios

L. Anido, M. Caeiro, M. Llamas, M.J. Fernández
Área de Ingeniería Telemática.
Depto. Tecnologías de las Comunicaciones
ETSI Telecomunicación, University of Vigo
Campus Universitario s/n, E-36200 Vigo, SPAIN
Ph: +34 986 812174, Fax: +986 812116
e-mail: lanido@ait. uvigo. es

Key words: Collaborative learning, courseware, CBT, ODL.

Abstract: This paper presents a Web-based tool aimed to provide a collaborative environment. Among others, its features allow any kind of data sharing on the Web, on-line synchronous communication among learners and instructors and the possibility to follow virtual presentations with an embedded Web-based slide projector. We used an electronic whiteboard resembling those which can be found in traditional classrooms to gather all this functionality:

1. INTRODUCTION

The use of multimedia material including text, audio, video and simulations offers an attractive environment to students, especially if the Internet is the framework used to support it. Learners are allowed to control what multimedia elements are delivered and when they are delivered using a structure of linked elements through which the user can navigate. This kind of structured multimedia environments are known as hypermedia systems and makes learners themselves the guides of their own learning (self-pace learning). This possibility is stressed by several authors [1, 2] who state that students can learn better and more quickly when they are stimulated by a high level of interactivity and if they can follow a personal way to learn. In particular, they can customise their participation to the courseware, establishing the amount of time to be spent on each topic and individualising their self-paced learning [1]. Thanks to Computer Based Training (CBT) the student is the new centre of the learning process, exploring the environment and building knowledge and skills [2].

Furthermore, conventional distance learning has played an important role in education because of its added value, especially for those combining learning and work. In conventional distance education systems, trainees are allowed to study anywhere, at any time and at their own pace since there are no timetables or schedules to follow. This seems to us, at first sight, an important feature for life long learning: the learner is released from tight rules which may help young students but not workers or other non-full-time students. Thus, important economical advantages are achieved, both for trainees and instructors and for the academic institutions that are responsible for carrying out the learning process. Since there is no need for travelling, trainees and tutors do not have to waste their time and money. Traditionally, conventional mail was used to deliver whatever printed material was needed by distance students. Fortunately, new technologies allow us to use the full capabilities of CBT in distance learning. Multimedia and Hypermedia material can be delivered to students by CD-ROM as personal computers have been brought into general use. Therefore, we are able to combine the advantages of both CBT and distance education systems.

On the other side, the universality of the World Wide Web [3,4] makes it a practical and suitable platform for delivering courses and other educational material. Web-based education offers remote access from everywhere and at any time. Because every lecturer can place educational material on the WWW, every student or trainee in the world, not only those close to that lecturer, is able to access that information. As a consequence, anybody with an Internet connection can take advantage of the knowledge of the best experts in every subject.

The WWW is, as we said before, an interesting virtual room where the learning process could be enhanced. Nevertheless, at the same time, it is a huge forest where those who are not used to Web-based learning could get lost. To avoid this, it is essential to provide learners with suitable tools to resemble what they can find in conventional education. According to Johnson [5] there are three different ways to carry out the learning process:

- Individualistic Learning.
- Competitive Learning.
- Cooperative Learning.

Conventional education copes with all of them. Teachers are able to choose the way they consider the best. Web-based distance learning systems can provide the first but also the other two ways. Networked systems provide communication channels among the different agents involved in the learning process. Over them, we should develop the appropriate tools to turn the Web into a real virtual classroom and actually benefit from their inherent potentiality.

The rest of this paper is organised as follows: the next section presents a tool for web-based collaborative learning. Section 3 shows its functionality as a standalone drawing tool. Then, in section 4, we present its use as a virtual slide projector over the Internet. Finally, some conclusions are presented.

2. A TOOL FOR WEB-BASED COLLABORATIVE LEARNING

We designed and developed a Web-based tool to provide distance learners on the Web with the needed collaborative atmosphere. A set of cooperative features have been embedded in a virtual electronic whiteboard whose functionality is presented thorough this section. This tool can be used as a helper application or it can be included in existing ODL systems with a slight effort.

The tool we present (see figure 1) replaces a conventional whiteboard in a Web-based educational environment. Our electronic whiteboard provides learners and instructors with a board where they can introduce multimedia information, from text and images to video. Whatever data any of the participants introduce would be synchronously presented to the others, in this way they share the same board. In order to identify the source for any data introduced in the board, every user is assigned a different colour when they enter the current whiteboard session. Thus, we are providing our trainees and instructors with a virtual shared space where they can share any graphical information as they were used to doing on traditional classroom whiteboards.

Because of the limited size of computer screens our whiteboard allows us to manage different panels or boards in the same session. A privileged user will be responsible for moving forward and backwards through the different documents or panels. The tool will assure that the same document is shown to every user involved in the active whiteboard session.

Learners and instructors could be invited to participate in any of the current active whiteboard sessions. At the same time, they could request by themselves to be accepted in a session whose topic (publicly available to every user) is interesting for them. When a new user is introduced in a given session, the document would be sent to the whiteboard to update its contents.

There exists a basic messaging system that can be configured to be used as a mail system or like a news bulletin system. The main aim of this messaging facility is to allow *off-line* information exchange and to provide a rendezvous system.

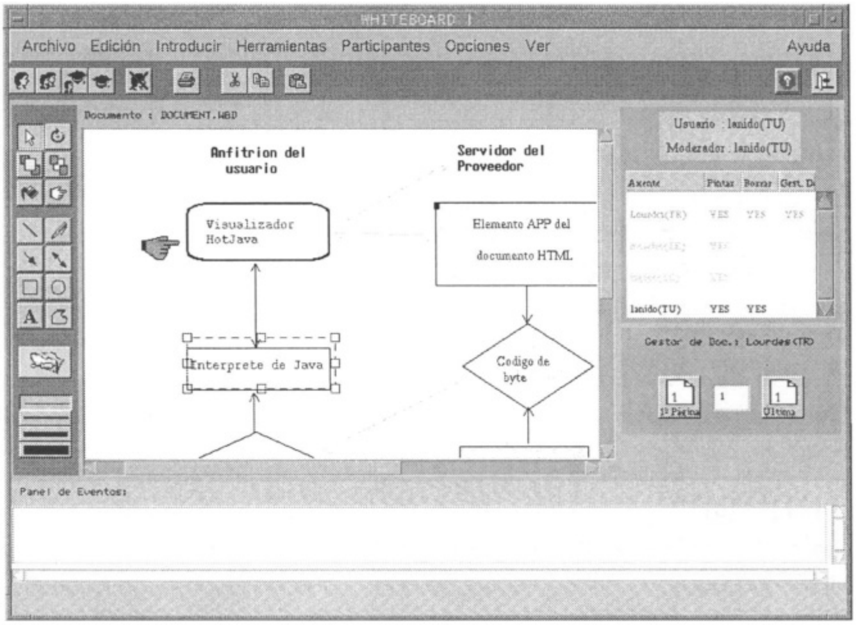


Figure 1. The user interface

The needed software to support the whole cooperative functionality of our tool can be delivered to the users' computers in the form of a Java [6] applet or application. Thanks to the use of Java computing it is possible to use our whiteboard on any computer regardless their architecture or operating system with no software set up. Therefore, whoever may want to use our software does not need to worry about configuration or compatibility issues. Even those who are not familiar with computers will be able to use the whiteboard.

Figure 1 shows the whiteboard user interface. In this example there are several people involved in the whiteboard session. They can communicate each other using graphical elements and text as if they where in a traditional classroom.

The student who begins a whiteboard session or the first teacher involved will act as its *chairperson.* The *chairperson* is responsible for the following duties:

– Inviting others to join the current collaborative session. The *chairperson* is responsible for the users management. He/she will receive requests from users to be included in his/her chaired session and he/she is also able to remove *participants* from it.
– Users' permissions management. Every user involved in a communication session could be allowed to introduce data and delete their own data, introduce data and delete data (including data introduced by others) or just to act as a listener. Users may request for a higher permission level to the *chairperson* who is in charge of the permissions management.
– Appointing the *document manager.* As we said before, it is possible to show different boards to the users involved in a given whiteboard session. The person responsible for deciding which document is shown at every moment is appointed by the *chairperson.* Any *participant* is allowed to apply for the *document manager* post.
– Assigning colours to *participants.* Whenever a new user gets involved in the whiteboard session, he or she will be assigned a different colour which will identify him or her in the current session. This colour can be changed by the *chairperson* and any *participant* is allowed to request the *chairperson* to change their assigned colour.

The *chairperson* or any *participant* appointed by him could act as the *document manager.* A document is just another panel or board shown on our whiteboard. In the same whiteboard session, there may exist a set of documents linked to it as a logical sequence although only one of them is shown at a given time. The *document manager* is responsible for:

– Including a new white document in the current whiteboard session. Every document is assigned a number which is used to create a logical sequence of panels, from the first one (number one) to the document inserted in the last position (number n). New documents may be added at the end of the sequence or between any two pre-existing panels (e.g. a document inserted between panels 2 and 3 becomes panel number 3, former panel number 3 changes to number 4 and so forth).
– Selecting the document to be shown to *participants.* The *document manager* is allowed to "move *participants"* to the next document, to the previous document or directly to the first or last document in the logical sequence.

Participants are those learners and instructors involved in a whiteboard collaboration session. Once a user is enrolled in a particular session, he or she will be allowed to introduce and delete (their own) data. This default access privileges can be changed by the *chairperson.*

Every *participant* is identified by a different colour which is shown in the participants' panel, where every user's privileges are also shown. Requests from users to change their current role in the whiteboard session (change their access privileges or apply for the *document manager* post) would be sent to the *chairperson* for approval. Of course, both the *chairperson* and the *document manager* are allowed to introduce and delete information as any other *participant.*

3. A STANDALONE DRAWING TOOL

The presented whiteboard gathers a collaborative educational environment to exchange information with advanced drawing tools. Apart from lines, arrows, and basic geometrical figures, there exists a predefined set of complex figures to be used by *participants*; among others these include flowchart-like figures and some predefined symbols. Every element introduced in the board is treated as an individual object which may be rotated, moved, resized, re-coloured, filled with a different colour, etc. In addition, those users with the right access privileges can group objects in order to be considered as a set for future operations. The purpose of these functions and elements is to provide an easy way to introduce information in the whiteboard. Regarding the introduction of text, this can be included in the main panel, like any other multimedia component, or it can be inserted in a separate frame dedicated to serve as a chat window. For the text included in the main frame the user has several options he or she may configure. *Participants* can also introduce images in different formats which would be embedded in the current panel and any other multimedia type whose content could be managed by the Web browser. In short, most outstanding features of powerful graphical applications were taken to a collaborative environment oriented towards Web-based learning.

The whiteboard can also be used as a standalone drawing tool (see figure 2). Design and development of whiteboard documents can be prepared *off-line* and, afterwards, be included as part of any whiteboard session. Individual multimedia elements, graphical objects, particular documents or a set of related documents can be stored at the learners' or the instructors' personal computers or at their personal accounts at server side. Thus, students and teachers are allowed to concentrate on the learning material at *on-line* time instead of wasting a high percentage of time in the development of the data to show to others.

4. USING THE WHITEBOARD AS A VIRTUAL SLIDE PROJECTOR

In the previous subsection we outlined the possibility to use our collaborative tool as a standalone application to develop information to be presented to others and which, hopefully, will be eventually transformed into knowledge. On the other hand, there exists the *document manager* role which is in charge of moving forwards and backwards the contents of the whiteboard through a set of related documents organised as a logical sequence. These two features can be combined to obtain a particularly interesting functionality for our collaborative tool: it can be easily used as a virtual slide projector where all the *participants* in a given whiteboard session will act as the audience and the *document manager* as the speaker. To improve the virtual projector capabilities of our educational tool we also allowed the embedding of HTML documents inside the board to ease the presentation development process and include any preexisting Web document in a given presentation.

Virtual speakers, usually instructors but not restricted to learners, should prepare their presentations using the whiteboard as a standalone application, organise their presentation into slides (whiteboard documents) and develop a logical sequence of documents which will constitute their eventual presentation. The developed presentation could be stored at their personal computer or at server side. Then, the speaker only needs to start or be included in a whiteboard session and become its *document manager*, introduce the developed documents in the current session and we are done! As the *document manager,* he/she would be allowed to move forward to the next document (slide) in the whiteboard (presentation) introducing additional information, if needed, in the chat area. At any point, usually when the presentation has finished, the *chairperson,* who may be the user himself/herself, can give the needed permission (allow speaking) to those *participants* who requested it (raised their hands to ask questions) and even make comments and suggestions over the speaker's presentation.

The implications of the presented functionality are obvious to those who try to make the WWW a suitable environment for teaching. We are providing them with a set of collaborative features embedded in an integrated environment where no assumptions about the particular learning contents or ODL platform were made.

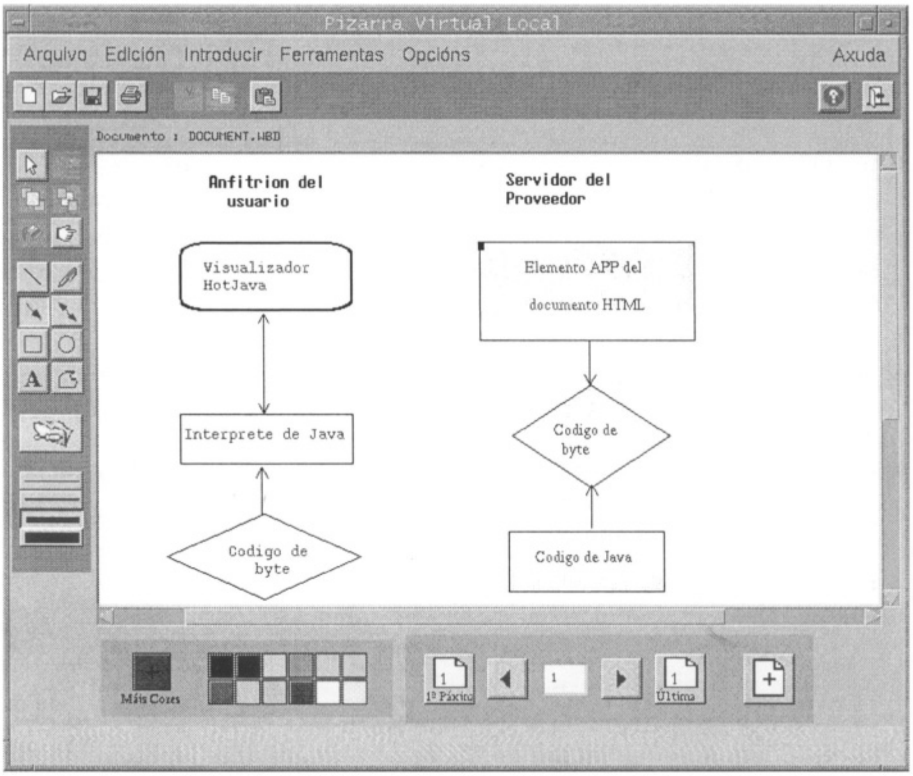


Figure 2. The whiteboard as a standalone drawing tool

5. CONCLUSIONS

WWW-based courseware may become a horrible headache for learners. Distance learning should not mean for students feeling alone, and this seems a real danger in Web-based training. A collaborative environment is absolutely essential to prevent trainees from learning alone.

In this paper we have presented a set of cooperative tools embedded in a Web-based virtual shared whiteboard. Our tool allows multimedia data sharing, synchronous and asynchronous communication, Web-based brainstorming and virtual presentations on the Web. It is aimed to adapt the idea of a classroom we may have in our mind to a Web environment and ease the learning process. In addition, since no assumptions were made about the learning contents or educational information system, it can be easily embedded in many existing Web-based learning systems with no additional software set-up by students or teachers. We do believe it complements many teleteaching environments to provide an appropriate framework for life long learning.

REFERENCES

1. M. Chirico, F. Giudici, A. Sappia and A.M. Scapolla. "The Real Experiment eXecution Approach to Networking Courseware". *IEEE Transactions on Education on CD,* Vol. 40, no 4, November 1997.
2. G. Da Bormida, D. Ponta and G. Donzellini. "Methodologies and Tools for Learning Digital Electronics". *IEEE Transactions on Education on CD,* Vol. 40, no 4, November 1997.
3. World Wide Web. http://www.w3.org
4. T. Bray. "Measuring the Web" *Fifth International World Wide Web Conference.* Paris, France. May 1996.
5. D.W Johnson and R.T. Johnson. *Learning Together and Alone: Cooperative, Competitive and Individualistic Learning.* Englewood Cliffs, N.J. Prentice-Hall 1997.
6. J. Gosling, B. Joy and G. Steele. *The Java Language Specification.* Addison-Wesley. ISBN 0-201-63451-1.

The Next Step in Computer Based Education: the Learning Technologies Standardisation.

L. Anido, M. Llamas, M.J. Fernández, J. Santos and M. Caeiro
Área de Ingeniería Telemática.
Depto. Tecnologías de las Comunicaciones
ETSI Telecomunicación, University of Vigo
Campus Universitario s/n, E-36200 Vigo, SPAIN
Ph: +34 986 812174, Fax: +986 812116
e-mail: lanido@ait.uvigo.es

Key words: CBT, interoperability, standardisation, re-use, standards, recommendations.

Abstract: The learning technologies standardisation process is taking the lead role in the research efforts into computer-based education. Institutions like the IEEE, the US Department of Defense and the European Commission have set up committees to deliver recommendations and specifications on this area in order to provide interoperability between different educational systems. This paper presents an up-to-date survey on this field.

1. INTRODUCTION

The increasing use of the Internet and its technological capabilities have allowed a high number of Internet-based distance learning platforms to come up. As they are usually developed *ad-hoc* to meet the requirements of a particular institution, heterogeneous systems appear with no interoperability mechanism among them. As a consequence, courses developed for a particular system cannot be reused by a different one. The students' performance profiles as stored by a given system cannot be read or managed by a different educational platform. Educational services provided by an institution cannot be easily shared with other learning environments, even if they agreed to do so, because of the heterogeneous technologies involved and proprietary solutions adopted.

There exist important efforts in the learning technologies standardisation process led by several institutions and projects. Their main aim is to contribute to commonly accepted standards to define learning data and metadata as well as recommendations for the development of software architectures devoted to computer-based education.

Much work has been done and is being done in the learning technologies standardisation area. Among the main contributors to this effort let us mention the IEEE's Learning Technologies Standardization Committee (LTSC) [1], the IMS Global Learning Consortium [2], the Aviation Industry CBT Committee (AICC) [3], the US Department of Defense's Advanced Distributed Learning (ADL) initiative [4], the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) project [5], Getting Educational Systems Talking Across Leading Edge Technologies (GESTALT) [6], the PROmoting Multimedia access to Education and Training in EUropean Society (PROMETEUS) [7], the European Committee for Standardization Information Society Standardization System, Learning Technologies Workshop (CEN/ISSS/LT) [8] and the Gateway to Educational Materials (GEM) project [9]. A list of the acronyms used through this paper is provided in Appendix A.

The IEEE's LTSC is the institution that is currently gathering recommendations and proposals from other learning standardisation institutions and projects. Specifications that have been approved by the IEEE go through a more rigorous process to become ANSI or ISO standards. In fact, a new ISO/IEC JTC1 Standards Committee for Learning Technologies, SC36 [10], was approved in November 1999. Below we present the main outcomes of these standardisation efforts obtained so far. The most outstanding results are from metadata for learning resources, definitions of learner records and profiles, formats for course structures and packages, formats for questions and tests and definitions of learning architectures and run time environments. This standardisation is being developed at the time of this writing and, therefore, this survey may encourage the reader to contribute to this process from his/her own experience.

2. METADATA FOR LEARNING RESOURCES

The learning metadata definition area has been one of the main focuses for the learning standardisation community during the last few years. Metadata is just data about data, in this case, data about educational data and resources (e.g. language used to write a specific course). The purposes of these definitions are, among others: to allow humans to search, evaluate, acquire and use learning objects, to enable sharing and exchanging of learning objects across any technology-supported learning system, to let computer agents to automatically and dynamically compose personalised lessons for an individual learner, to enable educational institutions to express educational content and performance standards in a standardised format that is independent of the content itself. In short, it aims at the standardisation of the description of learning resources.

Important outcomes have been already delivered. One of the main contributors to this effort is the IEEE LTSC's [1] Learning Objects Metadata (LOM) working group. The LOM specification [11], version 4 from February 2000, describes learning content cataloguing information. It specifies the syntax and semantics of learning object metadata, defined as the attributes required to fully and adequately describe a learning object. Relevant properties of learning objects include type of object, author, owner, terms of distribution, format, teaching or interaction style, grade, level, mastery level and prerequisites. The structured approach to metadata definition implies that the actual data elements of a learning resource are grouped into meaningful categories. The base LOM scheme consists of nine such categories: General, Lifecycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation and Classification.

LOM metadata is becoming a standard de-facto among the learner community. However, this specification is the result of the effort of many contributors. Among them, the European ARIADNE project [5] and the IMS project [2] stand out. ARIADNE uses LOM version 3.8, to which it contributed significantly, for indexing and exploiting its network of interconnected knowledge pools (KPS), see figure 1. The IMS Learning Resources Metadata Specifications [12] (August 1999) is directly based on the IEEE's LOM with some changes based on implementation testing and detailed document reviews by the IMS Technical Board, which will probably be incorporated into the IEEE specifications. The IMS metadata specification identifies a minimum set of IEEE metadata elements called the IMS core (19 out of 86 LOM elements). The remaining IEEE metadata elements form the IMS Standard Extension Library, SEL, (67 out of 86 LOM elements). The IMS has also completed a survey to identify taxonomies and vocabularies, which can be used as values for the defined metadata elements.

The DoD's ADL [4] Sharable Courseware Object Reference Model (SCORM) [13], January 2000, applies the IEEE/IMS definitions to the three components of the SCORM model: raw media, content and courses. It provides the link between general specifications and specific content model.

Other system that has extended the LOM definition is GESTALT [6] that delivered its own metadata specification: GEMSTONES [14]. The main extensions of LOM include external rights management and the improvement of the quality of service description. GEMSTONES are used by the GESTALT brokerage service to locate learning resources. The Gateway to Educational Materials (GEM) [15] system also provides a brokerage service based on extensions of Dublin Core [16] metadata.

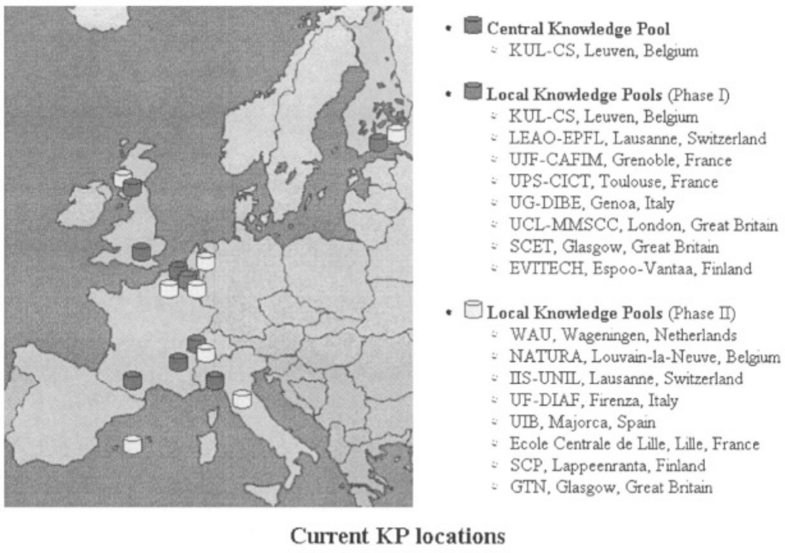

Central Knowledge Pool
KUL-CS, Leuven, Belgium
Local Knowledge Pools (Phase I)
KUL-CS, Leuven, Belgium
LEAO-EPFL, Lausanne, Switzerland
UJF-CAFIM, Grenoble, France
UPS-CICT, Toulouse, France
UG-DIBE, Genoa, Italy
UCL-MMSCC, London, Great Britain
SCET, Glasgow, Great Britain
EVITECH, Espoo-Vantaa, Finland
Local Knowledge Pools (Phase II)
WAU, Wageningen, Netherlands
NATURA, Louvain-la-Neuve, Belgium
IIS-UNIL, Lausanne, Switzerland
UF-DIAF, Firenza, Italy
UIB, Majorca, Spain
Ecole Centrale de Lille, Lille, France
SCP, Lappeenranta, Finland
GTN, Glasgow, Great Britain
Current KP locations

Figure 1 ARIADNE KPS Network

3. LEARNER RECORDS AND PROFILES

The description of learner profiles and records has also been studied in order to deliver recommendations on standards that allow the exchange of student data. The IEEE LTSC's [1] Public and Private Information (PAPI) specification [17], version 5 from January 1999, describes portable and implementation-independent learner records. Learner records are organised into four major categories: personal, preference, performance and portfolio information. They describe information about the learner, about his/her technical, learning and physical preferences, about the learner's history and about his/her current works. A particular file format to store student performance data was also defined by the AICC [3] as part of its guidelines for interoperability [18].

Based on PAPI, the IMS [2] Enterprise Data Model Specification [19] (January 2000) is aimed at administrative services that need to share data about learners, courses, performance, etc., across platforms, Operating Systems, user interfaces and so on. This data model is supported through the use of three data objects: person, group and group membership. The person object contains elements describing an individual of interest to the learning environment. The group object contains elements like a course instance, training programs, academic programs, clubs, courses, etc. The group membership object contains elements describing the membership of a person or group in a group. Group members may be instructors, learners, content developers, managers, mentors or administrators.

4. COURSE STRUCTURE FORMATS

The US Department of Defense ADL initiative [4], as part of its SCORM model [13], has identified an XML-based representation of a course structure format (CSF) that can be used to define all course elements, structure and external references necessary to move a course from one system to another. It is not course packaging, as the course structure is just one (albeit very important) of the elements needed to move a course from a given system to another one. CSF describes a course using three groups of information. The first group, called globalProperties, is the data about the overall course. The second, called block, defines the structure of the course, and the third group, objectives, defines a separate structure for learning objectives with references to course elements within the assignment structure. Course sequencing is defined using prerequisites and completion requirements for blocks, assignable units and objectives.

ADL's CSF is derived from the AICC [3] content model for course structures, properties, and objectives. The AICC in its guidelines for interoperability [18] (November 1999), has described a set of files for the basic data on the structure of a course. It includes all of the assignable units and blocks in the course. Its order in the file implies (but does not force) an order for presentation to the student. A personalised order is allowed using a fully specified table of prerequisites.

5. COURSE PACKAGING

The IMS [2] project is leading the standardisation process in this particular field. The IMS Content & Packaging Specification [20] (March

2000) makes it easier to create reusable learning resources. The key element is the package: an abstract description of a unit of reusable content. A package must provide all files and data needed to transfer the learning resources it embodies from one system to another. It is also possible to aggregate a package into a higher level one. The two components of a package are the manifest file and the physical resources, see figure 2. The manifest contains a metadata description of the package as a whole, one or more ways of organisation of the content, and can include or reference sub-manifests which describe the packages that have been included or referenced. The physical resources are a collection of resources physically included within a package.

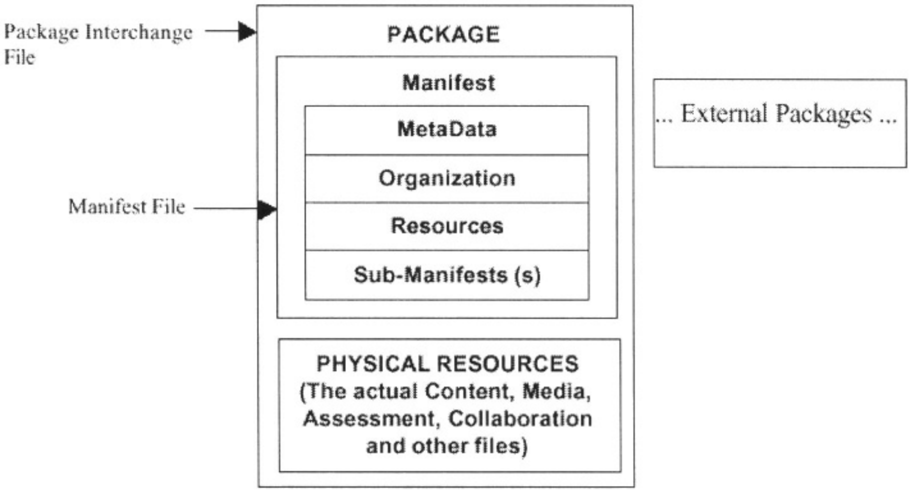


Figure 2: IMS packaging model file

6. QUESTIONS AND TESTS INTEROPERABILITY

In February 2000, the IMS project [2] has delivered the first specification on questions and test interoperability [21]. It addresses the need to share test items and other assessment tools across different systems, and describes the data structures needed to provide interoperability between questions and test systems, particularly those that are Internet-based. The key elements are assessments (basic test units), sections (containers for sections and items with a common testing objective) and items (the fundamental self-contained question/response block). The specification defines a taxonomy that describes a set of response types and different rendering forms.

7. LEARNING ARCHITECTURES AND RUN TIME ENVIRONMENTS

The IEEE LTSC's [1] Architecture and Reference Model working group delivered the Learning Technology Systems Architecture (LTSA), draft 5, in December 1999 [22]. The LTSA specification covers a wide range of learning systems. It is pedagogically neutral, content neutral, culturally neutral and platform neutral. Five refinement layers of architecture are specified, from the highest to the lowest levels as: learner and environment interactions, human-centred and pervasive features, system components (with four processes: deliverer, learning entity, evaluation, and coach; and two stores: learning resources and learner records), stakeholders perspectives and priorities and API's coding and protocols. They are applicable to a broad range of learning scenarios.

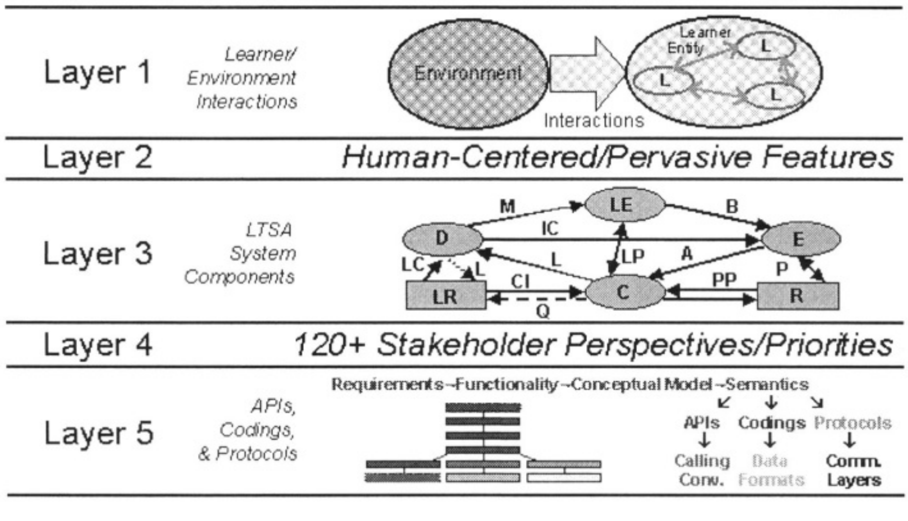

Layer 1
Learner/ Environment Interactions
Environment
Interactions
Learner Entity
L
L
L
Layer 2
Human-Centered/Pervasive Features
Layer 3
LTSA System Components
M
LE
B
D
IC
E
LC
L
LP
A
P
L
CI
PP
LR
C
R
Q
Layer 4
120+ Stakeholder Perspectives/Priorities
Layer 5
APIs, Codings, & Protocols
Requirements→Functionality→Conceptual Model→Semantics
APIs
Codings
Protocols
Calling Conv.
Data Formats
Comm. Layers

Figure 3: IEEE's Learning Technology Systems Architecture

Regarding the specification of concrete run time environments, the work by the AICC [3] and the DoD's ADL [4] stand out. The AICC guidelines for interoperability of Computer-Managed Instruction (CMI) systems [18] and the ADL's SCORM [12] based on the AICC specifications, deal with a common problem: in the past, authoring systems made the customer a captive of his/her own CMI system. In order to avoid this dependence between CMIs and CBTs (contents), a standard approach is defined to allow a single CMI system to initiate lessons from different CBT vendors. To accomplish this function, CMI and CBT must communicate by means of standard types of data: data from CMI to CBT to start the lesson, data from a CBT system to CMI needed to record student performance and assign the next learning unit, and data needed for evaluation of a lesson such as item response data, simulation performance data, etc.

Additionally, the GESTALT project [6] identifies a run time architecture made up of components from previous ACTS projects [23]: GAIA, Renaissance and Prospect. The system architecture comprises a learning environment, an administration system, an asset management system, a service for user profiles and a resource discovery service. Business objects comprising the interfaces among the various software components within the GESTALT architecture are identified. Middleware solutions were used for this purpose: DCOM [24] for interfaces among systems to be run in the same institution and CORBA [25] for those interfaces among systems from different institutions.

8. OTHER RECOMMENDATIONS

So far, we have presented those areas in the learning technologies standardisation efforts which have successfully delivered standards and/or recommendations. There exist other fields with incipient working groups or whose outcomes have not been published yet. Let us mention the work on unique student identifiers [1], Web-based learning [1, 3, 4], standards for user interfaces [3, 1] and recommendations for hardware and software platforms to support computer-based learning.

Other aspect we have to take into account is the global scope of computer/Internet-based education. Multi-nationality and multi-culture questions arise here. Organisations such as CEN/ISSS/LT [8] and PROMETEUS [7] work in close cooperation to identify requirements for standards-related activity in learning technologies bearing in mind the heterogeneous aspects, language and culture found in the European scope. As Europe is a multilingual environment, learner models must include the language that the learner understands in order to deliver contents appropriately. Other questions that appear as a consequence of the European multinationality are the establishment of a global policy for educational copyright and learner models. CEN/ISSS/LT has also delivered recommendations for technical specifications. For example, the metadata model they propose is mainly based on the IEEE LOM [11]. The strength of LOM for European Learning communities is its flexibility for adding local classification extensions. There already exist several European language translations of the LOM specification.

9. CONCLUSIONS

The heterogeneous nature of computer-based education systems led international organisations to face the need of standardisation. Standards and recommendations on learning technologies aim to provide a common framework to allow reusability and interoperability between vendors and consumers of computer-based learning material.

The International Standardization Organization (ISO) has recently set up a committee on learning technologies standardisation [10]. This committee is aimed at becoming the core sink for standardisation efforts currently led by different institutions. The IEEE's LTSC [1] stands out among them. LTSC standards are the result of the outcomes from other important efforts on several areas as presented in this paper. Some of the LTSC's results are becoming standards *de-facto*. For example, the LOM [11] metadata specification or LOM extensions are of common use to define properties for learning resources. It is expected that specifications on other aspects like learner records, run time environments or test definitions become wide spread enough to allow full interoperability among computer-based education environments. In the meantime, all of us are encouraged to contribute to this specification process.

REFERENCES

1. IEEE's LTSC website at http://ltsc.ieee.org
2. IMS Global Learning Consortium website at http://www.imsproject.org
3. Aviation Industry CBT Committee website at http://www.aicc.org
4. DoD's Advanced Distributed Learning website at http://www.adlnet.org
5. ARIADNE website at http://ariadne.unil.ch
6. GESTALT website at http://www.fdgroup.co.uk/gestalt
7. PROMETEUS website at http://prometeus.org
8. CEN/ISSS/LT website at http://www.cenorm.be/isss/Workshop/lt/
9. GEM project website at http://www.geminfo.org
10. ISO-IEC JTC1 SC36 website at http://www.jtclsc36.org
11. LOM Working Draft 4 at http://ltsc.ieee.org/doc/wgl2/LOM_WD4.PDF
12. IMS Learning Resources Metadata Specification website at http://www.imsproject.org/metadata
13. ADL's SCORM at http://www.adlnet.org/Scorm/scorm_index.cfm
14. GEMSTONES website at http://www.fdgroup.co.uk/gestalt/metdata.html
15. GEM project website at http://www.geminfo.org
16. Dublin Core metadata website at http://purl.org/dc/
17. PAPI specification, version 5.00, 1999-01-15 at http://edutool.com/papi/
18. AICC. CMI Guidelines for Interoperability at http://www.aicc.org/docs/tech/cmi001v301rtf.zip

19. The IMS Enterprise Profiles Specification, version 1.01, at http://www.imsproject.org/enterprise
20. The IMS Content & Packaging Specification, version 0.92, at http://www. imsproj ect.org/content
21. The IMS Question & Test Specification, version 1.0, at http://www. imsproject.org/question
22. IEEE's LTSC Learning Technology Systems Architecture (LTSA), draft 5, dated 1999-12-08 at http://edutool.com/ltsa/
23. ACTS projects website at http://www.infowin.org/ACTS/
24. Microsoft's Distributed COM (DCOM) website at http://www.microsoft.com/com/tech/dcom.asp
25. OMG's CORBA website at http://www.omg.org

APPENDIX A: ACRONYMS

ACTS.- Advanced Communications Technologies & Services.

ADL.- Advanced Distributed Learning. US Department of Defence initiative

AICC.- Aviation Industry CBT Committee.

ANSI.- American National Standards Institute.

ARIADNE.- Alliance of Remote Instructional Authoring and Distribution Networks for Europe.

CBT.- Computer-Based Training.

CEN/ISSS/LT.- European Committee for Standardization (CEN), Information Society Standardization System (ISSS), Learning Technologies Workshop (LT).

CMI.- Course Management Instruction.

CORBA.- Common Object Request Broker Architecture.

CSF.- Course Structure Format (by ADL)

DCOM.- Distributed Component Object Model (by Microsoft)

GEM.- Gateway to Educational Materials.

GEMSTONES.- Gestalt Extensions to Metadata Standards for ON-line Educational Systems.

GESTALT.- Getting Educational Systems Talking Across Leading Edge Technologies.

IEEE.- Institute for Electrical and Electronic Engineers.

IMS.- Instructional Management Systems Global Learning Consortium

ISO.- International Standardization Organization.

LOM.- Learning Object Metadata (by IEEE's LTSC).

LTSA.- Learning Technology Systems Architecture (by IEEE's LTSC).

LTSC.- Learning Technology Standards Committee (IEEE organisation).

PAPI.- Public and Private Information (by IEEE's LTSC).

PROMETEUS.- PROmoting Multimedia access to Education and Training in EUropean Society.

SCORM.- Sharable Courseware Object Reference Model

Foundations of Programming: a Teaching Improvement

M.V. Belmonte, C. Cotta, A.J. Fernández, I. Gómez, J.L Pastrana, J.A. Pedreira, F. Rus, E. Soler
Departamento de Lenguajes y Ciencias de la Computación, E.T.S.I.I., 29071 Teatinos, Málaga, Spain. E-mail: {mavi, ccottap, afdez, ivan, pastrana, pedre, rusman, esc}@lcc.uma.es

Key words: Programming, Multimedia, Teaching, Hypermedia.

Abstract: The ESPUMA project is a university project directed to the improvement of teaching in first-year subjects of programming. The ESPUMA project was born three years ago with the main aim of improving the quality of teaching and motivating the student to learn the foundations of programming. This project involves both the use of the new technologies in the classroom and the development of attractive graphical environments aimed at helping the student to study the topics of the subjects. In this paper we present the ESPUMA project and evaluate its three-year application by showing main results in terms of academic results and acceptance from the student's point of view.

1. INTRODUCTION

During the last decade, *Teaching* has emerged as one of the most exciting fields of research. The importance of teaching in society is well known. Evidence of the success of the research in the teaching field can be found in the increasing number of papers published on different subjects of teaching. One of these subjects is *Education in Computing* [1,2,3,4,5], which is the topic treated in this paper.

The School of Computer Studies (SCS) at the University of Málaga offers two degree-programmes (i.e., SCS-Management and SCS-Systems), each of which is a 3-year full-time course. The first academic year is very important to consolidate the basic concepts necessary to understand each topic of the full programme. *Foundations of Programming I* and *Foundations of Programming II*, denoted as *FP I* and *FP II* respectively, are two subjects corresponding to the first academic year in each of these degree-programmes. By the end of the year, the student is expected to be familiarised with the main notions of structured programming, to have some ideas on the concept of 'good style' in programming, and to understand the notions of abstract data types. Thus, the student should be able to write and modify small-scale programs using an imperative programming language including management and design of libraries. At the University of Málaga, these subjects are taught not only in the SCS but also in all remaining Engineering studies. This highlights the importance of these subjects [6].

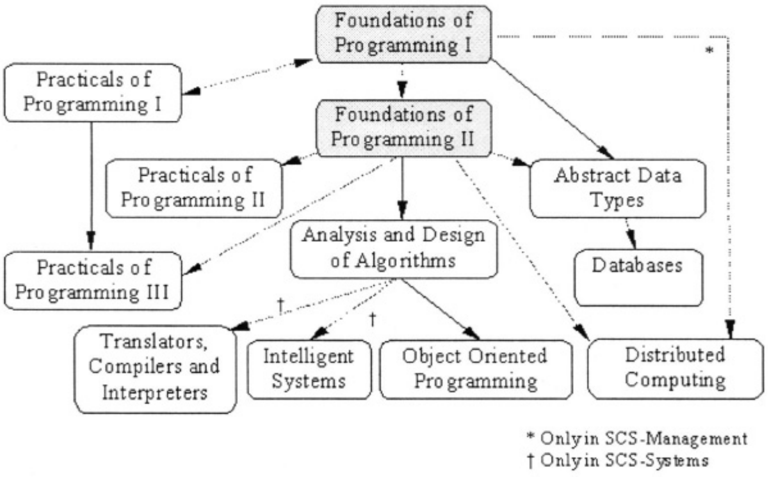

Foundations of Programming I
Practicals of Programming I
Foundations of Programming II
Practicals of Programming II
Abstract Data Types
Analysis and Design of Algorithms
Databases
Practicals of Programming III
Translators, Compilers and Interpreters
Intelligent Systems
Object Oriented Programming
Distributed Computing
*
†
†
* Only in SCS-Management
† Only in SCS-Systems

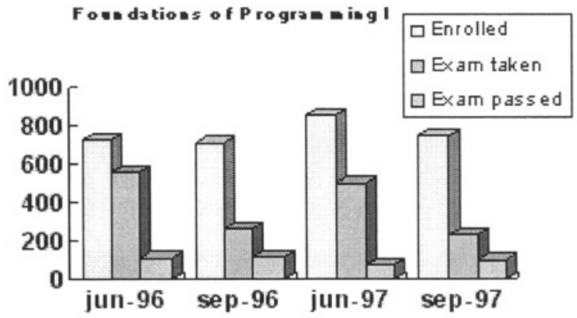
Figure 1. Subjects in the SCS depending on the topics taught in FP I and FP II

The correct learning of the concepts taught in FP I and FP II is essential to correctly understand the topics taught in further academic years (see Figure 1). The poor results obtained in these subjects in terms of the number of students taking and passing the final exams in the past years motivated the developing of the ESPUMA project. This project is led by a group of lecturers that are involved in the integration of the new technologies in the teaching field. The ESPUMA project was born three years ago with the aim of improving the teaching quality in the subjects FP I and FP II in the SCS at the University of Málaga. This project tries to complement the typical lectures of a lesson with new tools (resulting from the capacities provided by the Internet) that help the student to a better understanding of the topics of the subject.

In this paper we describe the ESPUMA project and show the main results obtained after three years of application.

2. ACADEMIC FRAMEWORK AND OBJECTIVES

As mentioned above, the subjects FP I and II are very important since they introduce the basic concepts necessary to complete the three year SCS course. The experience shows the existence of a high rate of academic failure for the first year students. Figure 2 shows the results corresponding to both June examination and September examination of these subjects in the academic years 95-96 and 96-97. Note that the rate of students passing the exams with respect to the number of students taking them is under the 28% and 45% in June and September examinations respectively.



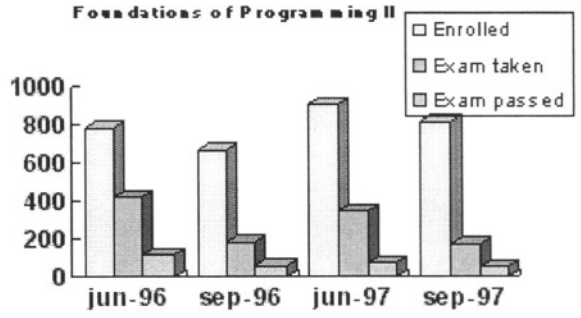


Figure 2. Academic results in FP I and FP II in the years 95-96 and 96-97.

Moreover, the problem of absenteeism among the students is worrying. Generalising, about the 57% and 71% of the students enrolled at these subjects do not take the exam in any of these examinations (June and September). This is a problem that must be considered seriously since, occasionally, the number of students attending a lesson is under the half of the enrolled students. Among others, we have detected some reasons for this:

- The problem of overcrowding in the classroom;
- Old-fashioned (and even obsolete) methods of teaching;
- Old-fashioned teaching material that has failed to move in the course of time.

Consequently, we considered that a renovation of the teaching programme as well as the teaching methodology was necessary. Then, the ESPUMA project was born in the academic year 1997-98 with the aims of *attracting the students* to the learning of the topics explained in the subjects FP I and FP II and of *increasing the quality* of teaching by:

1. increasing the <u>active participation</u> of the students during the lectures,
2. promoting the <u>attendance</u> of the students,
3. proposing <u>complementary tasks</u> to give the students the opportunity to work on concrete concepts and
4. supplying to the students new <u>software tools</u> that help them to mature the concepts acquired in the lectures.

It is important to mention that our project has managed to interest a lot of teachers and lecturers of other subjects and Engineering schools.

3. DESCRIPTION OF THE PROJECT

In this section we describe the ESPUMA project by identifying our two main proposals: (1) a new teaching methodology and (2) novelty software tools as well as additional material of teaching.

3.1 Methodology

In the traditional lessons, the lecturer explains concepts by using just a blackboard and/or slides to complement the explanation. These resources are valid for a lot of theoretical subjects but are insufficient in more practical subjects as those treated in this paper. Therefore, we propose the integration of the new technologies with the traditional lectures of a lesson. Specifically, we propose the use of *hypermedia systems* as an alternative to the current teaching methods. Hypermedia is defined as the science of the relations to structure, present and provide the users with direct accesses to the contents inside a concrete domain of information by making use of video, audio and text [7]. Hypermedia systems provide the typical elements necessary to motivate the students because they mean:

1. a powerful and flexible way to introduce concepts;
2. a way to construct tools with capacities to be extended;
3. a way to combine documents on different supports.

According to García Cabrera et al. [8], teaching must be open, multifunctional, co-operative and presented on multimedia environments. These are the main guidelines we have put into practice in the ESPUMA project.

3.2 Software Tools and Teaching Material

We consider that students need additional motivations to understand the concepts explained during the lessons. Thus, we have constructed, or are being constructed, a novelty set of software tools and teaching material to complement the explanations of the lectures.

3.2.1 Proposed Tools

We propose the following additional resources to be used in the teaching of FP I and FP II:

a) Attractive slides (see an example in Figure 3),
b) A development environment to program in a programming pseudo-language that helps the student to understand the global concepts of programming without particularising on a concrete programming language (see Figures 5 and 7).
c) An environment for executing interactive graphical animations led to a better understanding of the concepts acquired in the lectures (see Figures 4 and 5).

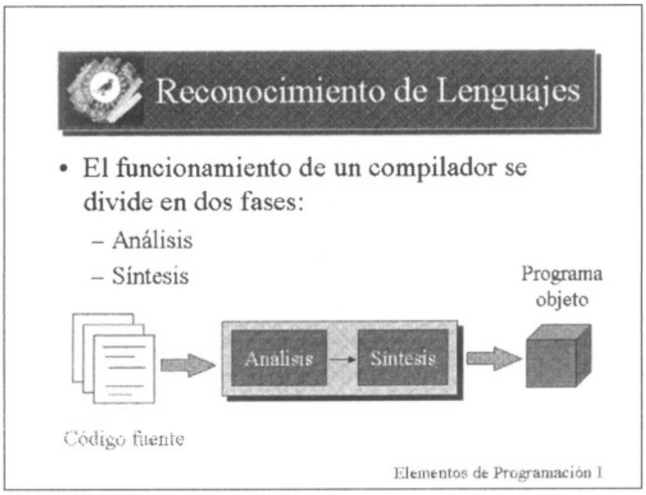
As complementary information, we also provide:

a) *A virtual class teacher:* an e-mail address is available for all the students enrolled in the subjects of FP I and FP II so that each student has a direct channel to ask doubts concerning to these subjects. Each query is firstly answered directly to the student who submitted it and later centralised and answered globally in form of FAQs (Frequently Asked Questions) in the web.
b) *A discussion forum* in form of newsgroup (relative to these subjects). This is under construction and we hope it will be working soon.
c) Administrative information relative to the subjects FP I and FP II on the web.

3.2.2 Current Services

In this section, we describe the services that are currently totally operative.

a) *Slides.* The main concepts and topics of the programmes of FP I and FP II have been summarised in a set of full-coloured slides revised by the lecturers teaching these subjects. This means that all the lecturers use the same slides. Also most relevant exercises have been summed up in these slides so that we get the students used to paying attention to the concepts and not to the writing of the exercises. The slides corresponding to each topic are available to the students at least one week in advance.



Reconocimiento de Lenguajes

- El funcionamiento de un compilador se divide en dos fases:
  - Análisis
  - Síntesis

Programa objeto
Análisis
Síntesis
Código fuente
Elementos de Programación I

Figure 3. Example of the new format of the slides

b) Virtual class teacher: the e-mail address *tutor-EP@lcc.uma.es* was created to answer the queries of the students.

c) Design of Graphical Interactive Tools: we have developed a set of graphical tools to visualise (graphically) concepts related to the subjects (e.g., pointers). These tools are graphical environments that interact with the student (see Figures 4 and 6). They have been developed in the Java programming language [9] and only require a navigator program to be used on the Internet.

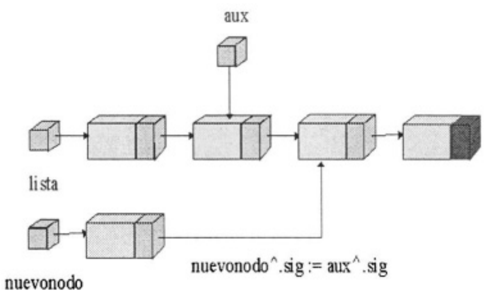


Figure 4. Graphical visualisation of inserting an element in a linked list

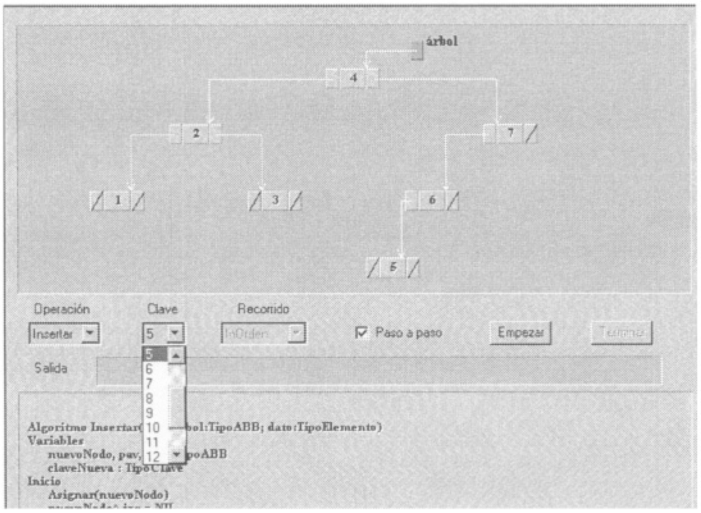
```
Pseudocódigo del algoritmo

Asignar (nuevonodo, Tamaño (Nodo))
(* Introducir información en nuevonodo *)
nuevonodo^.sig := NIL
SI lista = NIL ENTONCES
        lista := nuevonodo
EN OTRO CASO
        aux := lista
        MIENTRAS (aux <> NIL) AND
                 (aux^.info < nuevonodo^.info) HACER
                ant := aux;
                aux := aux^.sig
        FIN MIENTRAS
        SI aux <> NIL ENTONCES
                nuevonodo^.sig := aux^.sig
                aux^.sig := nuevonodo
        EN OTRO CASO
                ant^.sig := nuevonodo;
        FINSI
FINSI
```

Figure 5. Code in the programming pseudo-language (in Spanish) associated to the operation of inserting an element in a linked list

We also have the following services on trial:

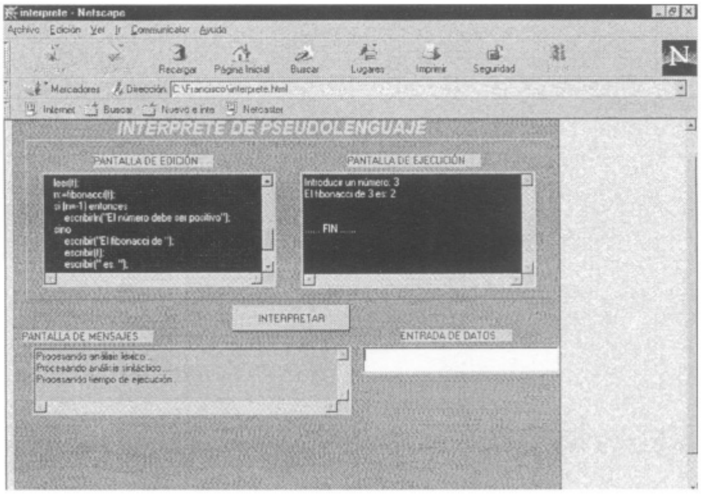
a) Tool to consult the marks in the web.
b) Tool to program in a programming pseudo-language (see Picture 7).

c) WWW address of the subjects. This address contains links to both the service of consulting marks and the services of complementary information.



```
Algoritmo Insertar(      bol:TipoABB; dato:TipoElemento)
Variables
    nuevoNodo, pav,        poABB
    claveNueva : TipoClave
Inicio
    Asignar(nuevoNodo)
```

Figure 6. Interactive program that shows binary search trees graphically.



```
leer(I);
n:=fibonacci(I);
si (n=-1) entonces
   escribirln("El número debe ser positivo");
sino
   escribir("El fibonacci de ");
   escribir(I);
   escribir(" es: ");
```

```
Introducir un número: 3
El fibonacci de 3 es: 2


...... FIN ......
```

```
Procesando análisis léxico...
Procesando análisis sintáctico...
Procesando tiempo de ejecución...
```

Figure 7. Tool for programming in the pseudo-language

4. RESULTS ANALYSED

The ESPUMA project is now three academic-years old and we have evaluated the results during this period.

4.1 Results compared

We have observed satisfactory results with respect to the number of students passing the exams. From our point of view this is very positive since it means that it is possible to motivate the students to improve their academic performance. Moreover, from the statistics we deduce that the decreasing tendency of students passing the exams is replaced for an increasing tendency. This seems to be result of the 'renewed' motivation of the students. Although the ESPUMA project is not totally completed since there are a lot of services on trial, the results encouraged us to continue it.

Also, the change of programme and teaching material is interesting for the students repeating the academic year since these attend the lessons as a novelty with a more practical approach. Moreover, the virtual class teacher service is a valuable help for these students since they do not have to attend all the lessons and can select the more interesting ones by asking via e-mail the contents of each lesson.

Figures 8 and 9 show that the rate of students passing the exams in the last academic years has increased significantly with respect to previous years. However, we are still worried about the high rate of students that do not take the exams.

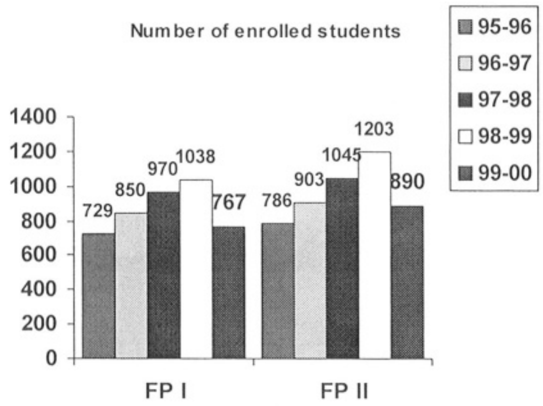


Figure 8. Number of enrolled students in FP I and FP II

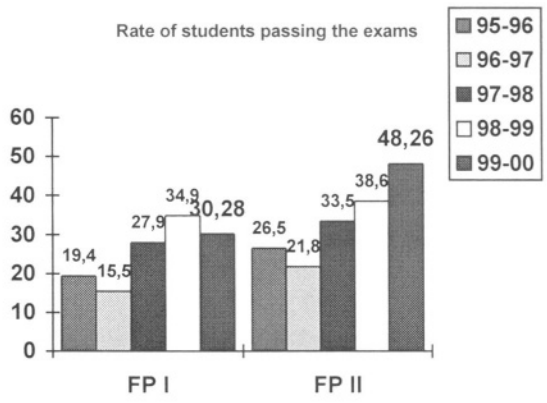


Figure 9. Percentage of students that pass the June exam with respect to the students taking the June exam in the last academic years in FP I and FP II

4.2 Student opinion

We are also interested in the opinion of our students so that this year we have written a questionnaire to know what they think about the current teaching methodology. The questionnaire consisted of 20 questions about different aspects of the subjects FP I and FPII such as the teaching ability of the lecturer, the material used in the subjects, the programme, topics explained, the additional material, software tools provided and other topics related. Each question was evaluated between 1 and 5 where:

- 1 corresponded to "I do not like it at all. It is very bad",
- 2 to "there is still room for improvement",
- 3 to "it is good",
- 4 to "it is very good" and
- 5 to "it is great!".

The overall evaluation was about 3.6. We interpret this mark as "quite good" and consider that the current material is appropriate for the lessons. This is the first year that we have evaluated this questionnaire and more years are necessary for a more complete evaluation. Anyway, the global mark near 3.6 encourages us to follow the ESPUMA project.

5. CONCLUSIONS AND FURTHER WORK

As future work we plan to add new material to avoid the decline of the interest of the student (mainly with respect to the students repeating the academic year). Also we are developing new software tools to help the student to a better understanding of the concepts explained in the subjects (i.e. new graphical simulations of theoretical explanations in the web).

We are also developing a tool to allow the student to self-evaluate. This tool would allow the students to evaluate their knowledge by means of theoretical queries and check their progress during the academic year.

Also to promote the interest of the students, we plan to organise seminars in which different groups of students will be encouraged to program more specific problems related with the subjects. Moreover, we plan to foster forums directed to the implementation of practical programs such as games. We think that these meetings are very attractive for the student since they can verify that the theoretical concepts acquired can be applied in practical problems. The aim is to make the student get a higher motivation to learn and not only to pass the examinations.

REFERENCES

1. Boyle, R.D. Specification and implementation of an `Active Learning' Facility. Tech Report TR 94.8, School of Computer Studies, University of Leeds, 1994.
2. Jenkins T. A Participative Approach to Teaching Programming. In Proceedings of the *6th Annual Conference on the Teaching of Computing / 3rd Annual Conference on Integrating Technology into Computer Science Education (ITiCSE '98)*, pp:125-129, Dublin, 1998.
3. Petre M. and Price, B. Teaching programming through paperless assignments: an empirical evaluation of instructor feedback. In Proceedings of *ACM SIGCSE/SIGCUE Conference on Introducing Technology into Computer Science (ITiCSE'97),* Uppsala, Sweden, 1997.
4. Siemer. J. The Computer and Classroom Teaching: Towards New Opportunities for Old Teachers. In Proceedings of the *6th European Conference on Information Systems*, Ed. W.R.J. Baets, pp:710-721, Aix-en-Provence, France, 1998.
5. Thomas P., Carswell L., Emms J., Petre M., Poniatowska B. and Price B. Distance Education over the Internet. In Proceedings of *SIGCSE/SIGCUE Conference on Introducing Technology into Computer Science (ITiCSE'96),* Barcelona, Spain, 1996.
6. Ureña, L. and García L. Análisis y valoración de la formación informática en los estudios universitarios. In Proceedings of *I Jornadas de Informática,* pp:475-484, Tenerife, 1995.
7. Bieber,M. and Isakowitz, T. Designing Hypermedia Applications. *Communications of the ACM,* 38 (8): 26-29, August 1995.
8. Garcia Cabrera, L. and Linarejos, M. Educatión e Hipermedia. In Proceedings of *II Jornadas de Informática,* Ed. B. Clares, pp:349-358, Granada, 1996.
9. Bishop J. Java, Elementos de Programación. Addison Wesley, 2ª Ed., 1998.

Development of Didactic Resources for Distance Learning based on Simulation.

F. Buendía García, J. V. Benlloch Dualde, J. M. Gómez Soriano.
Universidad Politécnica de Valencia, Escuela Universitaria de Informática, Departamento de Informática de Sistemas y Computadores.

Key words: Simulation, distance learning, hypermedia systems, Web technologies.

Abstract: Simulation is one of the most useful resources for teaching technical subjects. There are many examples of application to different engineering fields. As new information technologies based on the Web arise, simulation begins to be applied in distance learning environments. However, it is difficult and expensive to develop and evaluate didactic resources based on simulation in these environments. This paper proposes some key ideas to solve these problems and presents some examples related to the use of simulations to teach certain aspects about computer systems.

1. INTRODUCTION

Simulation is one of the most useful resources for teaching technical subjects. There are many examples of application to different engineering fields. In most cases, simulation tries to reproduce a real or fictitious situation that is very difficult to create or to control in a laboratory. In an educational environment, its use can enhance:

- The comprehension of the mechanisms that govern the simulated system.
- The ability to transfer the knowledge associated to the simulation.
- The motivation to improve the student's learning.

As new information technologies based on the Web arise, simulation begins to be applied in distance learning environments. In this context, the current work intends to provide a system for using simulations out of the traditional laboratory and without the direct support of the teacher. However, it is difficult and expensive to develop and evaluate this kind of didactic resources. This paper presents some key ideas to solve these problems:

- To reduce the development costs of simulation environments, by using concepts such as object programming and code re-use.
- To provide mechanisms to assist the students and to assess their progress by using simulations.

The impact of Web technologies has brought many simulation environments to the concept of “Web based simulation". Originally, these environments were based on CGI scripts connected to a server application which drives the simulation. If this simulation requires a more complex user interface, Java "applets" can be run in the browser client side. Currently, there are multiple library examples which contain general simulation functions based on Java such as *SimJava, JavaSim, Jsim, NetSim* [http://ms.ie.org/websim/survey/survey.html], C++ [http://www.cise.ufl.edu/~fishwick/simpack/simpack.html]), or specific domain [http://icosym-nt.cvut.cz/dynast/default.htm]. Using these libraries or other “ad hoc” tools, a big number of simulators have been built, representing a wide set of systems [http://www.cise.ufl.edu/~fishwick/websim.html].

These simulators have two main purposes: researching and teaching. In the teaching case, the educational role of most simulators is limited to reproduce, more or less exactly, the target system. This fact is a consequence of the use of simulation as a supplement of the teacher task in a traditional classroom or laboratory. The application of this kind of didactic resources in distance learning requires the development of new Web-based simulation environments which will contribute to integrate the use of simulation in a global learning process.

Two examples of such educational simulation environments are *Multiverse* [http://www.multiverse.co.uk/jtap/] and *CourseWare* [http://gingerbooth.com/courseware/index.html]. Both come from university groups which used the simulation resources in teaching or research activities. The predecessor of *Multiverse* was Interact [http://www-interact.eng.cam.ac.uk/]) which was oriented to engineering topics. On the other hand, *CourseWare* comes from a research group interested in simulation about ecology subjects. A common aspect in both environments is the separation between the development of the user interface and the simulation model itself. The main goal is to distinguish the roles of the programmer which is responsible for implementing the internal simulation details (simulation model) and the instructor which has to represent the interface of the simulation model in a right way for the student's learning.

Multiverse provides a visual tool for developing different user interfaces to access the same simulation model. The simulation model has to be implemented in Java language, using a predefined structure and certain function templates. This feature permits an easy configuration of the user interface to access the attributes and methods of the model. In this way, a non-programmer instructor can decide the presentation layout of the simulation and the data formats that are visible to the students. Several examples show the application of *Multiverse* tool in a Web context. They are structured in two parts: a description of the system to be explained, implemented as HTML text, and several user views to access the simulation model using Java applets. This educational approach can be very useful when it is used in an exploratory tutorial or there is an instructor who guides the simulation teaching. However, this approach is not adequate in a distance learning context. A first issue is the separation between the description of the system and the simulation of such system. In fact, both aspects are described in different pages without an explicit connection. Another question is the lack of evaluation mechanisms to check the use of the simulation by the student and the impact on his/her learning.

CourseWare represents a more integrated approach between the theoretical contents and the simulation model. It proposes a methodology to elaborate educational modules which contain certain contents using different media: text (e.g. HTML), graphics (e.g. GIF) and other simulations alike (Java applets). These modules are based on a configuration file in which the contents are specified as “readers” and their presentation and user interfaces are represented as “scenarios". This feature allows the instructor a higher freedom to include its own contents, as well as the way to present and connect them. Nevertheless, the role of programmer is still required when there are complex content such as simulations or multimedia animations.

The main advantage of the previous simulation environments is the separation between the roles of instructor and programmer to develop the different simulation aspects. The *CourseWare* solution presents a higher integration between these aspects and that is the starting point for the current work. The goal consists of using a formal model to describe the elements of an educational simulator and the relationships with other contents. These contents also include evaluation mechanisms that can be connected with the simulations.

The remainder of the paper is organised as follows: the next section presents the design of educational simulation resources based on the use of formal method such as a hypermedia model. The third section describes some implementation issues and an example application. The last section includes conclusions and further works.

2. DESIGN OF EDUCATIONAL SIMULATION RESOURCES.

Multiverse and *Couseware* are two examples of environments oriented towards the development of Web-based simulators with an educational purpose. However, they lack certain features to take full advantage of simulations in distance learning situations. Distance learning requires to set up learning sequences in which the simulation is considered as another educational resource. Each one of these sequences can use the simulation resources in a different way in order to provide a certain learning level. This requirement demands the use of a formal notation to define the different educational resources and the multiple ways to access them or to set up relationships between them.

The set of educational resources, the paths to navigate through them and their presentation aspects can be represented using a hypermedia model. This model has been used to represent complex information structures with multiple relationships. Some examples are the *Amsterdam* model [1] or the *Labyrinth* model [2]. The *Labyrinth* model has been applied in distance learning environments [3] and it is used as the reference model in the current work. In *Labyrinth*, a hypermedia application is represented by means of a **Basic Hyperdocument.** In addition, each user or group of users can have a **Personalised Hyperdocument** in which the users can adapt or modify the components of the Basic Hyperdocument to their own requirements (e.g. the student's knowledge in order to get an individualised learning). The Basic Hyperdocument is composed by the following elements:

- **Node** as a container of information to structure the set of educational resources.
- **Content** that represents the piece of information associated to each resource.
- **Anchor** as a reference locus into a node or a content and **Links** to relate anchors of different resources.
- **Events** to specify the access to a certain resource or how this reacts and
- **Attributes** to assign properties to the previous elements.

However, there is no mention in the model to the use of didactic resources such as simulations. The idea is to represent the simulation components as active contents that can be grouped into a node entity. It requires to define a special type of content (program) and the units that compose this content (program components). These program components can represent a set of simulator attributes or methods. Therefore, anchors can be assigned to these components and this feature allows links to set up connections between some of these components. In this way, an event triggered by a user who is browsing the description of a problem, can activate a certain simulator operation. Once the user has finished the browsing of the problem, the simulation results can be available.

In this case, the link is not only a navigational mechanism but it also requires a storage capability. For example, this capability can be used by the anchor source to send input data or to receive data in the anchor target. Figure 1 shows an example of hyperdocument in which a Simulator node is connected to other nodes. A Problem node can describe a complex topic which requires the access to a simulation to give an intuitive view of the problem. The Simulator node can also be accessed from a Test node to check some related questions or to access to a Presentation node to display the simulation results.

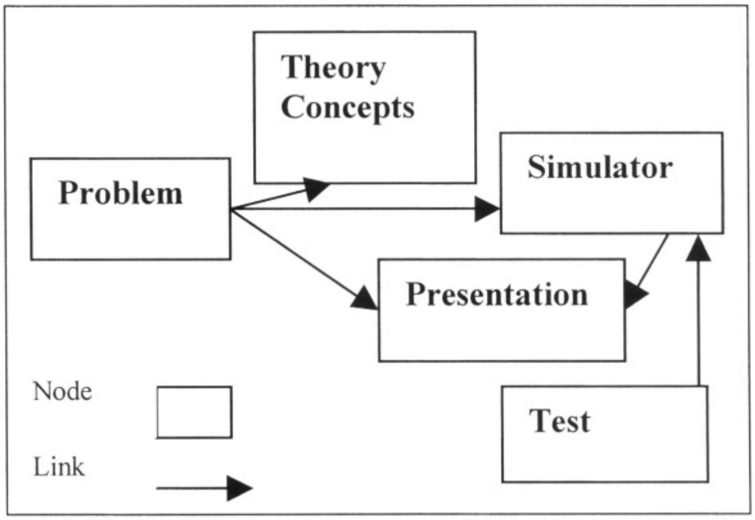


Figure 1. Hyperdocument structure example.

An aspect which remains unchanged with respect to environments such as *Multiverse* or *CourseWare* is the separation between the roles of instructor and programmer in the development of didactic resources. In the current work, a XML approach is used to allow the instructor to design the structure of the educational modules and to build certain types of contents by himself/herself. This approach is limited to passive contents such as texts, graphics, or other multimedia objects and it permits to publish these contents in different Web formats (not only HTML). Using this approach, the instructor can define the structure of an educational module [4], based on a set of nodes such as Problem Description, Prerequisites, Theoretical Concepts, References, Tests, and so on.

However, the development of simulators or other active contents is still a task assigned to programming experts. Environments such as *Multiverse* o *CourseWare* propose a common simulator access pattern to ease the implementation of user interfaces. This common pattern benefits from the characteristics of a language such Java to define abstract classes that can be adapted for a particular simulator. The same idea of common pattern is used in the current work to define the set of attributes and methods that forms the simulator interface. The simulation attributes are identified as:

- Control parameters.
- Input variables.
- Output variables.

And the methods are grouped as:

- Reading/Writing of the simulator control parameters.
- Writing of input data.
- Reading of output (result) data.
- Execution operations such start, stop, resume, pause, rewind, forward.

The first group of methods permits the access to the simulator control parameters, to read their values or to write new ones. These parameters represent the attributes of the simulation model and they are independent of the input or output data. The data input is performed using the next group of methods. Data output methods can be configured to produce different versions of the simulation results (e.g. text, audio, or graphic data). The last set of methods represents the control of the simulator operation. The user can request a global execution to obtain the final results or to execute steps in order to analyse the simulation progress.

The remainder of simulator components can be developed by means of any of the design methodologies commonly used. An example of object oriented programming method is applied in Buendía et al. [5].

3. IMPLEMENTATION AND SYSTEM EXAMPLE

Currently, there are few tools to develop didactic resources based on the previous design criteria. The option chosen in this work consists of building a prototype using standard Web technologies such as HTML, Javascript, and Java. The main idea is to use HTML as the basic format for publishing hyperdocuments and Java as the programming language for simulators. The links between the entities developed with these languages are based on the use of Javascript. Javascript is a script language similar to others like Perl, Phyton, ... Its main features related to the current work are:

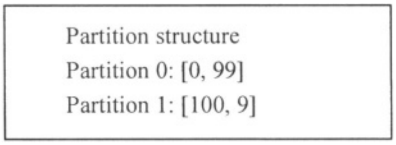
- Object programming model.
- WWW client functions such as validation and presentation data.

- Access to public attributes and methods, defined in Java applets.

These features allow Javascript to set up relationships between HTML and Java elements. This property is useful to implement links between contents. This is one of the design criteria previously defined in the hypermedia model. Such links have also the capability to transmit information using Javascript objects, which are accessible by the simulator, developed as a set of Java applets. Next, an example is used to show their application. The prototype can be accessed at the address [http://www.eui.upv.es/ineit-mucon/Applets/GesMem/GesMemApplet.html]. It has the next components:

- An HTML document that describes a computer topic such as the memory management and, more exactly, the algorithms to assign this computer resource using a distribution of fixed partitions. This document represents the Problem node in Figure 1.
- A Simulator node based on two Java "applets": one called *GesMemlnterface* that represents a user interface to access simulation operations from an external window, and another one, *GesMemEngine*, which represents the simulation model.
- A Javascript code embedded into the HTML document which includes the functions to access the simulation methods. Each function can be activated using a HTML button. The input and result data of the simulation are represented within a form element identified as *GesMemAccess.*

The problem presented in the example consists of analysing the procedures to manage a computer resource like the main memory between a set of computer processes. These procedures are limited to a memory model based on fixed partitions. The simulation control parameters that characterises the memory structure (e.g. total memory size or partition arrangement) can be configured, by accessing the attributes of the Simulator node. For example, the form shows the next data when the *Configuration* button is clicked.



```
Partition structure
Partition 0: [0, 99]
Partition 1: [100, 9]
```

The function assigned to this button is called *configure()* and its Javascript code is the following:

```
function configure(string){
  // applet verification
  control  =  document.GesMemApplet;
  if (control == null)  {
    Packages.java.lang.System.out.println("Denied
access");
    return;
   }
   else {
    Packages.java.lang.System.out.println("Granted
access");
  }
  // Access to applet configuration function
  control.motor.setValueString("Url",
control.ventana.url);
   control.motor.setValueString("Particiones",
string);
  // Data reading
  control.ventana.textoParticiones=
control.motor.getTextoDatos("Particiones") ;
  // Data output
window.document.PART1.texto.value=
control.ventana.textoParticiones. elementoDatos(0) ;
...
}
```

Such function receives a string argument which contains the configuration information used in this problem. This information is used to access the simulation method *setValueString* defined in the *GesMemEngine* applet. This method will set up the control parameters that configure the memory partition arrangement. Using the method called *getTextoDatos,* a representation of these data is displayed on the *GesMemAccess* form.

Once the memory structure has been configured, the next task consists of selecting the process set which represents the input data to the simulator. The selection of different set of processes with their own characteristics will permit to analyse the memory management in several scenarios. Such selection is performed by means of the Javascript function *load()* whose implementation is similar to the function *configure()*, previously described. Next, clicking the *Execution* button will start the memory management simulator with a certain configuration formed by empty memory partitions and a set of process which will be;assigned to these partitions. Using the *Results* button, the final assignment of processes to partitions is displayed. In a "step by step" mode, the partial results about this allocation can be obtained. The *GesMemInterface* applet provides the access to these "step by step" operations. The last button called *Correction* is an example how the simulator can be used to evaluate a given question. In this case, the question is about the percentage of memory fragmentation and the user has to answer it, using the *Results* information provided by the simulator.

4. CONCLUSIONS

The current work shows the educational possibilities associated to the use of simulations and the importance of integrating the simulation activities together with the theoretical contents. The potential of spreading the evaluation is remarked not only to assess the learning of theoretical concepts but also to include simulation aspects to check the student's ability against dynamic scenarios. After showing these possibilities, the problem is the lack of methodologies and tools to develop this kind of didactic resources on a large scale. This paper presents the use of a hypermedia model to ease the design of educational simulators and the integration with other didactic resources. Further works will intend to develop tools in order to translate hypermedia specifications into simulation implementations.

REFERENCES

1. L. Hardman, D.C.A. Bulterman, and G. van Rossum. The Amsterdam Hypermedia Model:Extending Hypertext to Support REAL Multimedia (technical report CS-R9306), Jan. 1993.
2. P.Díaz, I. Aedo y F. Panetsos. Labyrinth, an abstract model for hypermedia applications.Description of its static components. Information Systems. 22 (8), pp. 447-464, 1997.
3. A. López-Rey, F.Panetsos, M. Castro, J. Peire. Utilización del modelo Labyrinth en el diseño de la aplicación Now-meta. Congreso Nacional de Informática Educativa (CONIED'99)", 17-19 Nov. 1999, Puertollano, Ciudad Real.
4. F. Buendía, M. Quesada. Diseño de módules educativos accesibles por Internet para la enseñanza de sistemas informáticos. Congreso Nacional deInformática Educativa (CONIED'99)", 17-19 Nov. 1999, Puertollano, Ciudad Real.
5. F. Buendía, M. López, I. Blesa, J.V. Benlloch. Using Hypermedia Techniques for Developing Object-Oriented CourseWareabout Computer Systems. EAEEIE' 97 Annual Conference on Education in Electrical and Information Engineering. Jun. 1997 Edimburgh.

Tutormap

Past, Present and Future of a Mathematics Tutoring System

R. Criado, A. Gallinari, J.A. Velázquez
Escuela Superior de CC. Experimentales y Tecnología
Universidad Rey Juan Carlos
28933 Móstoles, Madrid, Spain
{r.criado,a.gallinari,a.velazquez}@escet.urjc.es

Key words: Mathematics education, mathematical computer system, Maple, tutoring system, interactive assessment

Abstract: Tutormap is a tool that extends the mathematical computer system Maple with educational facilities for tutoring and assessment. It allows the student to test his/her understanding and mastering of both the Mathematics presented and the Maple language. It also provides the teacher with means to prepare tests and automatically grade them. This mathematics tutor is designed to operate in a networked environment, so that students can work independently on their assignments. We present and discuss the working version of Tutormap, its future extensions and our experience with it.

1. INTRODUCTION

We teach introductory Mathematics courses to large-enrollment classes at our university, which is of very recent creation and offers interdisciplinary programs to future computer scientists and technicians. We need to meet educational and practical needs of students and faculty members. On the one hand, classes are formed by an audience that has different backgrounds, expectations and levels of knowledge. On the other hand, as faculty members, we needed to find a common methodology and ways to share, collect and organise our specific points of view, teaching styles and main goals.

Since the foundation of our university, three years ago, we decided to broaden the traditional setting of regular classroom and recitation sections of courses such as Linear Algebra, Calculus, and Discrete Mathematics. The use of computers would offer our students the opportunity to review and visualise general concepts and more specific elementary mathematical proofs, to improve their problem solving skills and to explore different paths that can lead to a complete answer to the suggested problem. Consequently, we decided to employ a symbolic algebra system in our computer laboratory classes [1].

We decided to provide the students with a system they could use not only in presence of their teacher (allowing them to receive and give immediate feedback), but also individually or in a collaborative learning environment outside the regularly scheduled classes. Therefore, they would be able to practise more on their weak points and to self-evaluate their own work. It was particularly important to give our students the opportunity to study independently because it improves their time-managing skills and allows the teacher to use more flexible means. Now, we can more easily tutor and make a more adequate selection of assignments, especially for those students who work outside the university or who already possess previous knowledge of some of the topics of the course.

Two important issues in the selection of a system were the lack of an adequate textbook and the usual time constraints imposed by the class schedule. We selected the computer algebra system Maple because it allowed us to achieve such issues. Lectures could be prepared by means of worksheets, which are interactive and reusable files that allow making computations, managing complex expressions and describing the problem-solving process. The user-friendly nature of the system helps to deal with time constraints: given that worksheets are handled as files in word processors, it is not necessary to learn a programming language to be able to employ it.

In general, Maple allows making operations of symbolic computation and numeric calculus, drawing graphics and developing programs. Other features of Maple are:

- The user can typewrite any expression that the system evaluates, and then it shows its result and waits for new inputs. Its syntax is similar to the one commonly employed in Mathematics, making the writing of mathematical expressions or functions more intuitive.
- It is an open and extensible system. The user-friendly nature of Maple does not imply a loss in computing capabilities since, if necessary, a programming language similar to Pascal or C is available and the user can extend the predefined system.

Maple is being used by us and by other universities (e.g. see [2]), but it is currently more adequate for professional mathematicians than for teachers or students. In the paper we describe Tutormap, a system that extends Maple with educational facilities. Tutormap is described in the second section of the paper. In the third section we describe our experience using it. Section 4 compares Tutormap with other systems. Finally, we summarise our conclusions and identify opportunities for future work.

2. DESCRIPTION OF TUTORMAP

Tutormap is an application that extends Maple with educational facilities. It runs on any personal computer equipped with Windows 98 or higher.

The teacher can use Tutormap with Maple to teach any subject in Mathematics, since he/she has to update only the contents of the corresponding lesson (by updating several Maple worksheets). The content of the lecture can be about either some Mathematics

or the proper Maple system. The teacher is given a tool to prepare and automatically assess them.

The availability of the full power of Maple for students while solving problems fosters “learning by inquiry and practice”. In addition, by taking tests, students can check on their understanding of the subject matter. The system is designed to operate both in standalone and in networked computers. In the latter case, students can work independently on their assignments. The system can later assess them and record the grades in a central database.

What follows is a description of the main features of versions 1.0 and 2.0 of Tutormap. The former is a fully working version [3], whilst the latter is under construction.

2.1 Tutormap 1.0

When the student starts a session, Tutormap offers a menu that allows choosing among several activities, as Fig. 1 illustrates. Upon the selection of any of the original options, Maple starts and the session will be switching between Maple and Tutormap, as the activity requires. In particular, the student may work on a Maple worksheet, following the instructions written by its designer. The student can also freely switch between Maple and Tutormap during an assessment. In this way, he/she can either approach each exercise following the given order (by solving it and then entering the corresponding answer) or postponing it.

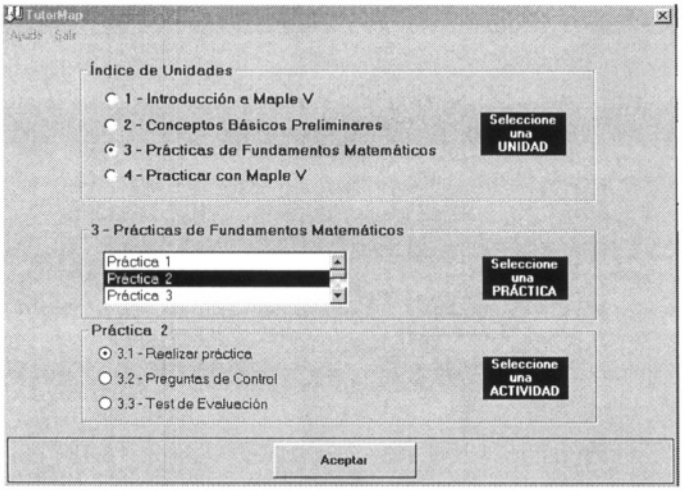


Figure 1 A screen snapshot showing Tutormap options

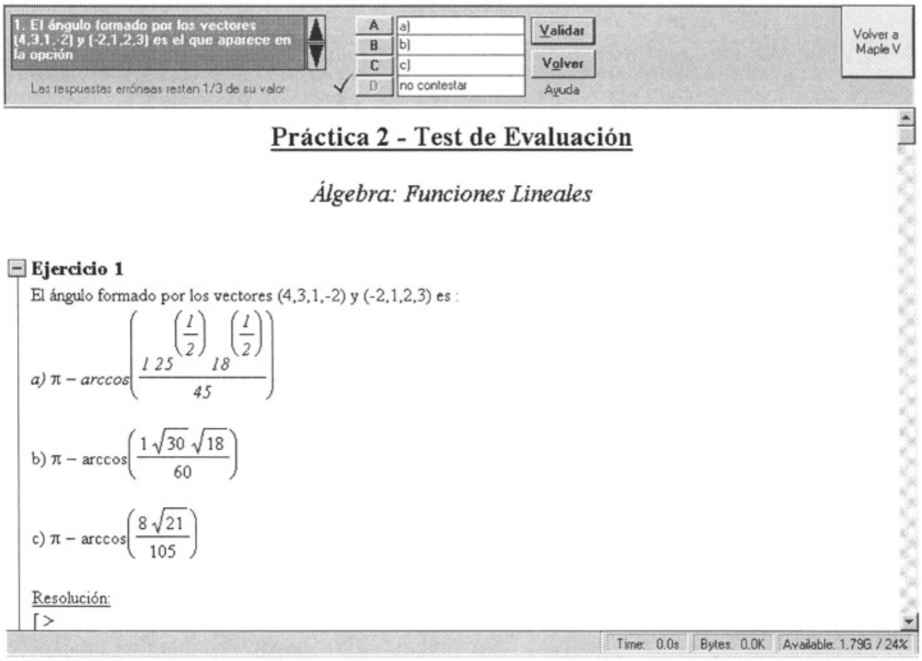
A key feature of Tutormap is that it gives full support to the student for reviewing, at any moment, all the prerequisites necessary to successfully answer the questions of tests. Prerequisites can be specific Maple commands, preliminary concepts of the course or concepts of the current lesson. Such prerequisite structure is in the form of options in the first menu of Fig. 1.

The understanding of lessons dedicated to learn Maple or prerequisites and the correct solving of assignments are assessed by quizzes containing multiple choice questions. When the student has answered all the questions, Tutormap offers the possibility of either reviewing the answers (sequentially) or finishing the examination. In the latter case, a summary with the number of successes and failures is given to the student. Tutormap also allows the teacher to store and maintain the grades obtained by the students in a database.

Fig. 2 shows a question in the upper part of the screen, and a worksheet describing an assignment in the rest of the screen.

2.2 Tutormap 2.0

Tutormap is currently being updated to run the latest release of Maple, Maple 6. As a consequence, the sections of the tutorial related with Maple syntax will now refer to version 6, and worksheets created with previous versions will be automatically updated. We are also making it more straightforward for teachers to modify the content of courses and all the related databases of the tutor.



1. El ángulo formado por los vectores (4,3,1,-2) y (-2,1,2,3) es el que aparece en la opción

Las respuestas erróneas restan 1/3 de su valor

Práctica 2 - Test de Evaluación

Álgebra: Funciones Lineales

Ejercicio 1

El ángulo formado por los vectores (4,3,1,-2) y (-2,1,2,3) es :

a) $\pi - \arccos\left(\frac{1\; 25^{\left(\frac{1}{2}\right)}\; 18^{\left(\frac{1}{2}\right)}}{45}\right)$

b) $\pi - \arccos\left(\frac{1\sqrt{30}\sqrt{18}}{60}\right)$

c) $\pi - \arccos\left(\frac{8\sqrt{21}}{105}\right)$

Resolución:

[>

Figure 2. A screen shot simultaneously showing a Maple worksheet and a Tutormap question

Originally created as a support tool for regular classroom teaching, our tutor was employed during laboratory assignments in presence of an instructor who could answer questions and guide students. Because of this, the system seems to lack more interactive, adaptive and complete features that would foster independent study. To this end, new chapters will be added with a change of style. In particular, more examples and problems will be included to better balance the rate between theory and practice. At the same time, the characteristics of quizzes will be modified in order to replace multiple choice questions with a more complex assessment style, based on a guide for the student through the inquiry and evaluation of different resolution paths.

One of the most relevant changes is that the user will not longer be constrained to work with Maple algebra computer system, since it will be possible to make use of other applications, such as Splus or Mathematica (in time, the list of applications available will be longer).

The next big step is to connect Tutormap to the Web. That can be used to extend the potential educational impact of the system substantially. The students could connect remotely and have a human tutor answering their questions and grading their work. More important, all the information available on the Web could become part of the learning process. Finally, to implement Tutormap on Java would allow access to remote databases of lessons, assignments and tests.

3. EXPERIENCE

Tutormap was introduced to first year Computer Science students at the Universidad Rey Juan Carlos during the academic course 1997/98. It was used in quarterly courses on Linear Algebra, Discrete Mathematics and Calculus. Some of the topics developed were linear spaces and functions, matrices, coding theory, limits, differential and integral calculus, differential equations, and an introduction to numerical methods in linear algebra.

The system Tutormap was most effective when we used it as a tool for motivating learning rather than as an assessment tool. Although all the students had a personal computer at their disposal, they began either to compete or to collaborate with each other. They often created work teams, and exchanged and discussed their results. If they did not obtain the highest grade, they would go through the assignment as many times as necessary, trying to correct their mistakes. Students even asked the teacher to make more assignments available before laboratory time, so that they could become familiar with the problems they would be asked to solve. A significant part of them experienced with the system in their spare time.

One of the questions that were part of the official teaching evaluation for the course "Algebra I" (Spring 1998) was to express the degree of agreement, from 1 (strong disagreement) to 7 (total agreement), with the following statement: "The material provided during the laboratory classes is appropriate". A total of 70 students, who used the system Tutormap as the only tool during their laboratory classes, answered as follows:

Table 1

1	2	3	4	5	6	7
0	0	4	5	13	30	18

A total of 102 students answered a similar question included at the official teaching evaluation for the course "Calculus I" (Autumn 1999). In that case the results were the following:

Table 2

1	2	3	4	5	6	7
3	9	8	19	22	30	11

At the end of each term, the students took an exam where they had to use Maple to (individually) solve ten problems similar in difficulty to those presented during the course. Grades were exceptionally high. For instance, of 119 students who took that exam in Algebra (Spring 99), 107 of them got a passing grade. In a similar exam for the course of Calculus I (Autumn 99), the passing rate was of 92 over 106 students. In all the courses in which we use Tutormap, more than 70% of the students passed that exam.

4. RELATED SYSTEMS

There exists a large variety of systems that share relevant features with Tutormap, and most of them were created to give an answer to problems analogous to the ones we faced. We will simply comment on some recent developments.

A Web-based system is incorporated into the calculus sequence at the Embry-Riddle Aeronautical University [4]. A Web page was created in order to give students all the information related to their program. In particular, worksheets containing daily scripts for the class, references, exercises for the students to work on in teams are included for the calculus classes. In addition, students can take quizzes with Maple, work in groups and take team exams.

Introducing tools for self-evaluation and interactive tests into the teaching methodology gives positive educational results (see, for instance, the study completed in the context of the Technology Enhanced Secondary Science Instruction (TESSI) project [5]), so they are becoming a common element of many mathematics education systems.

Most assessment systems are evolving to more complex learning tools, where emphasis is put on the use of tutoring systems and adaptive presentation technology.

The Web-based system OWL developed for General Chemistry at the University of Massachusetts [6] was created as an electronic quiz/homework grader providing corrected answers and informative feedback. Now, it is being used in 9 departments, including Mathematics. It contains a database that collects permanent records of the progress of the students who are using the system to complete their assignments. Present and future extensions of OWL include the introduction of intelligent tutors and more interactive learning environments.

The Web-based adaptive instruction system Arthur [7] is employed at the University of Cincinnati. By means of multimedia technologies, it makes several different teaching styles available to the students and guides them to find the most effective for their learning process. Before being able to go from a topic to the next one or to switch to a new teaching style, the user must take a quiz that can be graded by instructors or by the system itself.

The interactive problem solver MathEdu [8] is built on top of the Mathematica system and it is based on the principle of "programming by example". It guides the student through the resolution of a problem by suggesting different paths. It can be used in mathematics courses to combine theory and practice.

A tendency to an intensive use of the Internet can also be seen. An example is the Qwhiz Web game [9] for Mathematics and Science. It is an Internet tool offered by NASA's Learning Technologies Project that allows the user to select a difficulty level and play against the computer or other contestants.

All of these systems, including Tutormap, are works in progress. The ultimate goal is to create an environment that could adapt to the specific needs of the student and of the teacher, offer an interactive learning methodology, enhance critical thinking and simplify the access to the huge amount of information made available by new technologies.

5. CONCLUSIONS

We have described the main features of the system Tutormap. It is a complement to Maple that provides educational support for tutoring and grading. It provides a comprehensive prerequisite structure and it allows flexible interaction with Maple.

We have also described the positive results obtained by using Tutormap in our computer laboratory sections. We consider that there are two key reasons in its success. The first one is that, as the system employs a computer algebra system, students find support for problem solving activities, such as considering alternative approaches, experimenting, guessing, testing and analysing results. Second, the neutral interactive grading of mathematical assignments with quizzes was essential on providing motivation and a challenge to students. In summary, we believe that the educational principle "learning by inquiry and practice" has been achieved.

We have also explained present and future improvements of our system. Tutormap 2.0 will be a general tutoring and assessing system, not constrained to Maple. We also plan to develop a new version of Tutormap for the Web in order to make it available from remote locations and to allow the user to take advantage of Web resources.

ACKNOWLEDGEMENTS

This work was partially supported by the Comunidad Autónoma de Madrid under project no. 07T/0036/98.

REFERENCES

1. F. Ballesteros, A. Bujosa, R. Criado and L. J. Martín, "The impact of mathematical computation systems in mathematics education" (in Spanish), *II Jornadas Nacionales de Innovacíon en las Enseñanzas de las Ingenierías – Comunicaciones (volumen II),* Madrid, Spain, December 1996, pp. 861-867
2. J. Krone, "Using symbolic computation for teaching data structures and algorithm analysis", *SIGCSE Bulletin,* vol. 28, n. 4, December 1996, pp. 19-24 and 32
3. R. Criado Herrero and J.A. Velázquez Iturbide, "A tutor/corrector for mathematics education based on mathematical computer systems", to appear in *SIGCUE Outlook.*
4. John R. Watret, "Moving into the 21st century with high tech/high touch teaching: Integrating the curriculum on-line", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 187-192
5. D. Eichorn, J. Woodrow, "Self monitoring: A study of student interactive assessment", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 193-199
6. D. Hart, B. Woolf, R. Day, B. Botch and W. Vining, "OWL: An Integrated Web-based Learning Environment", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 106-112
7. J. E. Gilbert, C. Y. Han, "Arthur: An adaptive instruction system based on learning styles", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 100-105
8. F. Díez and R. Moriyón, "Doing mathematics with MathEdu", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 247-251
9. S. Smith, "Qwhiz: An interactive tool for Math and Science learning", *Proceedings of the International Conference on Mathematics/Science Education and Technology* (*M/SET 99*), pp. 166-170

Using teachers as heuristics evaluators of educational software interfaces

E. R. de Almeida Valiati*, M. Levacov**, J. Valdeni de Lima***, M. Soares Pimenta***
**Instituto de Ciências Exatas e Geociências, Universidade de Passo Fundo - Brazil*
evaliati@inf.ufrgs.br
***Instituto Latino Americano de Estudos AvançadosUniversidade Federal do Rio Grande do Sul- Brazil*
mlevacov@penta.ufrgs.br
****Instituto de Informática, Universidade Federal do Rio Grande do Sul - Brazil*
valdeni@inf.ufrgs.br, mpimenta@inf.ufrgs.br

Key words: interface project, educational software, usability, heuristic evaluation

Abstract: The process of usability evaluation demands the use of several techniques in order to facilitate acquisition of differentiated and expressive data regarding efficiency, easiness in using and interface learning. In some particular types of interface, as in the case of educational software, the process also requests the participation not only of interface experts, but also expert evaluators in the domain of the knowledge involved in the project. This article presents the results of an experiment involving teachers and interface experts, as heuristics evaluators of an educational software.

1. INTRODUCTION

Lately, as computers are becoming not only an indispensable resource for work in offices, but as means of communication, entertainment and education, the area of Human Computer Interface has had its importance recognised. With the development of personal computers and their growing expansion and evolution [1], the need for development of highly interactive applications (with increased complexity and richness), grows in volume, therefore the issue of usability becomes vital for the success of a product.

However, in educational software, those issues become still more important and justified, because of the obvious implications that usability problems can cause in this type of application, where a total friendly and transparent interface is desired, as to the educational goals intended by the use of a computer as an educational tool and resource.

It is unquestionable that interface projects demand an interdisciplinary team [2], and a user centred methodology should be adopted, in which the evaluation process, the central point, could be considered among the different apprenticeships of the development [3].

Several studies, reported in [2], [3] and [4], have recognised the benefits of interdisciplinary and collaborative work and, mainly, the users' involvement in the evaluation process, due to the diversity of knowledge and experiences demanded by projects of interactive systems.

This article is based on our experience application of heuristic evaluation techniques, in the evaluation of the usability of educational software, using teachers and interface experts as evaluators. The second section presents the technique and the aspects considered to facilitate its use to evaluators who are not expert in interfaces. The third section describes the object of interface evaluation. The fourth section reports and analyses the obtained results. The fifth section presents the final conclusions.

2. HEURISTIC EVALUATION

Among the methodologies used in interface projects, the technique of heuristic evaluation refers to the most informal method, easy and fast to be applied [5], because it basically consists of the use of a small group of evaluators (usability experts) that examine the interface and judge its conformity with established usability principles (called the "heuristics").

Jacob Nielsen [6] suggests a group of ten basic heuristics of usability that should be followed by evaluators in the sessions of usability inspection of the interface. These are listed below:

1. Visibility of system status
2. Matching between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetics and minimalist design

9. Helping users recognise, diagnose, and recover from errors
10. Help and documentation

In this technique, the evaluators can be experts in usability, consultants in software development with special knowledge (for example, in a particular style of interface), final users with knowledge of contents or tasks, or other professional users. The final heuristic evaluation is based on combining inspection reports from a set of these independent evaluators to form the list of usability problems.

For this experiment five evaluators were involved, two of them were experts in interfaces and three were non-experts, an educator and two geography teachers with domain expertise on the topics covered by the educational software evaluated.

In the sessions accomplished by interface experts, the heuristics and procedures proposed by Jacob Nielsen in [6,7] were used literally. However, for the sessions involving the non-experts, the procedure was:

a) In order to facilitate the understanding of the evaluators' task and to enhance their performance the heuristics were adapted as follows:
   Heuristics (criteria) of Usability, to be appraised:

1. Matching between the system and the real world
– Does the software speak the students' language, that is to say, does the information is in an accessible, familiar language for them, are not only the text but also the icons, images and all the other elements of the interface designed to be easy to understand?

2. User control and freedom
– Is the user free to choose what he or she wants to do in the software? That is to say, does the user have total control of the resources offered by the software, as well as the pace of presentation of information or the actions he or she will perform? For example: to undo an operation, to redo it, to leave a screen and to come back to another, to quit the software at any moment, to repeat a given information or situation, etc.

3. Consistency and standards
– Does the software keep a clear standardised presentation pattern of the information among its different screens, that is to say, is there a deliberate consistency in the way information is presented or in the way the resources are made available for the student's use? Does it enable the student to guess that different objects and actions might have similar meanings or behaviours?

4. Error prevention
– Does the software contain few execution mistakes and when there is one it supplies the user with information which prevents the user from making incorrect operations or taking precipitate decisions. Does the software

design uses mechanisms that ensure easy understanding and manipulation, helping students during the interaction and avoiding operation mistakes?

5. *Recognition rather than recall*
– *Does the software supply (making it visible) information that allows the student to recognise objects, actions, and options, instead of having to remember abstract instructions?*
6. *Gratifying and subjective satisfaction*
– *Is the software attractive, pleasant, and aesthetically captivating? Do texts and dialogues contain irrelevant and unnecessary information, so that the interaction becomes tiresome or annoying? This is important, because it is well known that extra information (or media) competes with more important information and promotes the decrease of the student's attention span or interest for learning.*
7. *Helping users recognise, diagnose, and recover from errors*
– *When necessary, is the user supplied with error messages, expressed in plain language, which help diagnose the problem, and constructively suggest a solution?*
8. *Help and documentation*
– *Does the software supply a manual or documentation that explains how it should be used? Does it possess a help resource that can be accessed to any moment? If it is there, is this material clear, accessible, and simple to use?*
9. *The software is simple and natural*
– *Are images, icons, texts, expressions and other available resources simple to use and to learn? Can they be naturally accessed, becoming a transparent interface for the student, so that the content and the abilities to be developed are clearly more important than the software in itself?*
10. *Flexibility and efficiency*
– *Does the software supply resources that allow multiple ways of exchanging information? Does the student have the possibility to execute multiple tasks as well as access multiple channels of information (auditory, visual, etc.)? Is the software efficient in its operation and in the supplied resources, in order to make its educational purposes possible?*

b) As Nielsen suggested, heuristic evaluation is performed by having each individual evaluator inspect the interface alone. A copy of the adapted heuristics, a detailed explanation of each criteria with examples from other software and instructions on how to report the usability problems found were given to the evaluators.
c) After a brief training session, each evaluator alone (in order to ensure an independent and unbiased report) tested the heuristics and the usability

issues in the design of the interface of all screens. For each detected problem, the evaluator described his/her observations and considerations to an observer (normally called the "experimenter"), who wrote them faithfully down in the appropriate form. Immediately after that, the evaluator revised the transcription and gave that problem a degree of severity according to Table 1, below:

Table 1: Severity scale attributed to problems detected in the test of heuristic evaluation

Severity	Meaning
1	Problem that does not influence or harm the execution of a task, the understanding of the content or the development of the expected abilities.
2	Problem that has little influence, but does not prevent or harms the student in the execution of a task, the understanding of the content or the development of the expected abilities.
3	Problem that affects the execution of a task, the understanding of the content or the development of the expected abilities slightly.
4	Problem that disturbs and hinders the student sensibly in the execution of a task, the understanding of the content or the development of the expected abilities.
5	Problem that severely harms or prevents the execution of a task, the understanding of the content or the development of the expected abilities.

3. OBJECT OF EVALUATION

This experiment's object was the educational software “BRAZILIAN GEOGRAPHY” - module I. This software belongs to the informative hypertext/hypermedia type, aimed at 6th grade students at elementary school. The subject of this particular module is physical geography of Brazil including information on climate, vegetation, hydrography, etc.

According to Figure 1, the screens of the interface in that module basically present the following structure:

1. identification area, in the upper right-hand corner of the screen, with the title and the icon of that particular sub-module;
2. menu of sub-modules, in the upper left part of the screen, with icons in inactive state;
3. area of options, on the left-hand side of the screen, with icons such as map, menu, game and sound;
4. menu, in the bottom part in the left-hand corner of the screen, with themes related to the sub-module being studied;

5. area of information, in the middle of the screen, where texts and images are presented;
6. navigational area, in the bottom right-hand corner of the screen, with right and left arrows for backward and forward buttons.

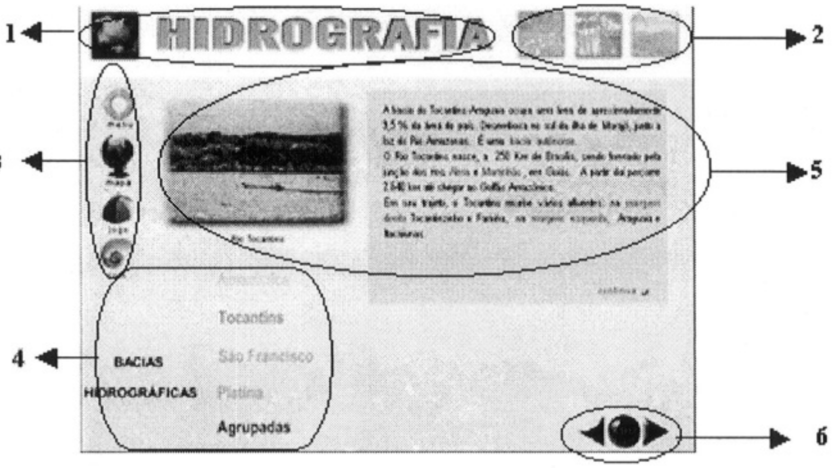


Figure 1 Screen example of the software on Brazilian Geography

4. RESULTS OF THE EVALUATIONS

Sessions had variable length among the evaluators' groups, since a fixed time was not defined. Teachers evaluators accomplished an average of 75 minutes, and interface specialists made an average of 55 minutes.

In this experiment, the obtained results were analysed taking in consideration both the domain of the problems and their number: 1) by more than one evaluator (of both groups); 2) only by the interface specialists; 3) only by teachers; and 4) by all the evaluators (adding the subtotals obtained in items 1, 2 and 3).

Two domains were related to the problems found by both groups: 1) specific problems of usability, such as, "*the sound icon seems to be active but nothing happens when a user clicks it*"; and 2) problems related to contents, such as, "*In [vegetation], there is a grammar mistake in the third paragraph of the text on prairies*," or "*In [hydrographic], the images presented on 'bacia Platina' have little relationship with the text on the same subject*".

After getting rid of duplicated problems, what remained was classified and distributed in the following way, as described below, in Table 2:

Table 2. Classification and distribution of the problems found by evaluators

Detected Problems:	**Problems of the type:**	
	Presentation of the contents	**Specific of Interfaces**
for teachers and interfaces experts	6	21
Subtotals	6	21
only for teachers		
P1	9	10
P2	1	1
P3	3	4
Subtotals	13	15
only for interfaces experts in		
E1	0	16
E2	3	10
Subtotals	3	26
Totals	22	62

Analysing Table 2, it is easy to verify that the subtotal of specific interface problems was larger among the interface experts, although teacher P1 has detected the same number of problems in this domain in relationship to experts E2. As expected, another difference could be observed in the content analysis, in which teachers were responsible in detecting a subtotal of 13 problems of this type against only 3 problems detected by the interface experts.

Another interesting data to be considered, although not represented in the table, relates to problems found by more than one assessor, that is to say, problems that presented duplicity in occurrences. Here, from 6 problems of content presentation, 67% of them were only duplicated among teachers' evaluations. 71% of the 21 interface problems found were duplicated only among the interface experts' evaluations.

This data led to the conclusion that, in spite of the evaluators of both groups being capable of detecting problems in both domains, each group of evaluators obtained better performance in its area of expertise.

However, analysing the teacher's performance as heuristics evaluators, a significant contribution by evaluators/teachers in the total of detected problems can be found. From the 84 problems detected at all, 27 problems were found by more than one evaluator (of both groups), 28 problems were found only by the teachers and 29 problems were just found only by the interface experts, as can be seen in Figure 2:

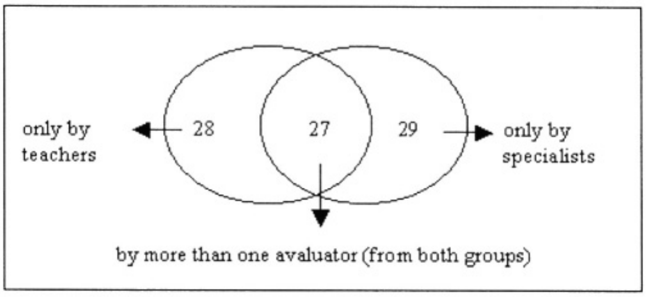


Figure 2. Total number of detected problems

Another data that allowed us to verify teachers' potential as heuristics evaluators of educational software was the total of problems detected by each evaluator, before discarding the superimposition of the detected problems among the evaluators, as displayed in Figure 3:

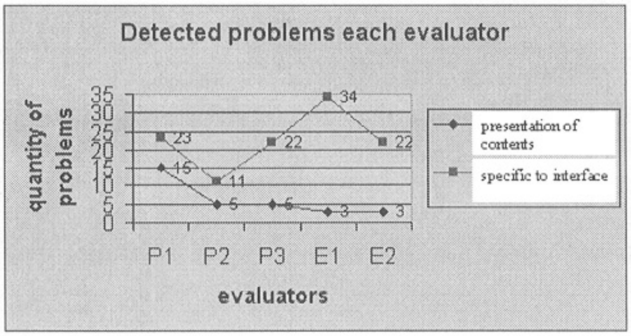


Figure 3. Amount of detected problems

As Figure 3 shows, through the use of the heuristic evaluation technique, teachers were capable to find a significant amount of problems in both domains, as compared to the amount of problems detected only by interface experts, who were able to detect only 3 problems each.

5. FINAL CONCLUSIONS

The expressive results obtained with this experiment prove that teachers are important as potential collaborators in the process of evaluating educational software because, besides being capable of finding specific interface problems (in a relatively high average, compared to the knowledge level that they possess in this area), they are irreplaceable in the detection of problems related to the presentation of contents, an area that is fundamental for this type of application.

Those results show, therefore, how easy the learning and application of the evaluation method is, as well as the validity of a training phase and of the adaptation of the heuristics to the evaluators/teachers. It is important to stress that such heuristics can be reused in future evaluations as a form of checking its real efficiency and aid in the evaluation of educational software with non-specialist evaluators.

Besides the amount of identified problems, the teachers supplied better qualitatively information by describing the problems, not just by telling the quantity of times the problem occur, but, its impact, the sceneries where it could happen and suggestions for possible solutions, which makes these reports very important for the elaboration of interaction practices with users.

The only disadvantage presented in the evaluations accomplished by each teacher was the significant number of reports about the occurrence of a same problem and the variety of problems identified in a same report, what demanded a posterior analysis of those reports, for the correct score of the problems indeed identified by each assessor.

Another data point that deserves attention refers to the high index of severity of the problems detected by both evaluators, reaching the average of 79% of the problems with degrees of severity from 4 to 5, which checks, according to experiments told in [8], that the method of heuristic evaluation allows the identification of a high number of problems of serious gravity.

Even so, in spite of the excellent performance presented by the teachers in the test sessions, it is noticed that the specialists' participation in interface is indispensable in the detection referring to usability problems. Thus, it is observed that each group of evaluators cope with specific problems of each domain and both contributions should be used in a complementary way, for a more efficient evaluation.

REFERENCES

1. Dix, A. et al. Human-computer interaction. Prentice Hall Europe, 1998.

2. Preece, J. et al. Human-computer interaction. Addison-Wesley, 1995.
3. Hix, D., Hartson, H.R. Developing User Interfaces. John Wiley & Sons Inc. New York, 1993.
4. Sommerville, I. et al. Sociologists can be Surprisingly Useful in Interactive Systems Design. Research report: CSCW/1/1992, University of Lancaster, 1992.
5. Nielsen, J. Guerilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. Available at http://www.useit.com/papers/guerrilla_hci.html, 1997.
6. Nielsen, J., Mack, R. L. Usability Inspection Methods. John Wiley & Sons Inc. New York, 1994.
7. Nielsen, J. How to Conduct a Heuristic Evaluation. Available at http://www.useit.com/papers/heuristic/heuristic_evaluation.html
8. Jeffries, Robin et al. User interface evaluation in the real world: a comparison of four techniques. In: CHI'91 Human Factors in Computing Systems, 1991.

Test Construction and Management with Network Adaptive Control

J.A. Díaz Villalta, A. Neira, A. Alguero, J.A.L. Brugos
Universidad de Oviedo, Departamento deInformática.
Campus de Viesque 33271 Gijón (Asturias) Spain.
E-mail neira@correo.uniovi.es Tlf. +34+ 985182481 Fax: +34+ 985181986

Key words: Distance Learning, Adaptive Testing, Bayesian Test, Intelligent Tutoring System

Abstract: Testing by computer is becoming more and more important in the present Tutoring Systems and Distance Learning Models. In this work, starting from several previous contributions and considering the assessment methodologies specified mainly by the Institute of Education Sciences of the University of Oviedo, an integrated model of test processing is proposed consisting of the following elements: an open items editor compatible with the current word processors that permits the highest flexibility at the moment of composing the enunciations and distracters with different multimedia elements, an items classification system in thematic areas making sorting and content balancing feasible in test working and facilitating the portability in different environments, a mechanism of selection and presentation to the student that permits simultaneously teacher predefined tests, a random access to the items for training and a self-assessment bayesian system, and finally, by means of a Client/Server architecture, an application for net management that facilitates their use as much in local area networks (LANs) as extended through the Internet.

1. INTRODUCTION

In the current academic environment, the interest that tests are having is increasingly higher in order to evaluate the students' knowledge acquisition. They are equally used as methods for assessing knowledge at a general level, as first filters to access related concepts in the study progress or as a study mechanism for training, learning and verification of knowledge comprehension and assimilation, developed by the students themselves.

In recent years, because of the implementation of new technologies in virtually all the academic areas, the computers have been made available to the majority of students, both for performing tasks and teaching the subjects to be studied. This fact, united to the proliferation of distance tutoring systems and other commercial models for teaching by computer, joined on the development of LANs and the widespread use of the Internet, gives us the possibility and necessity of fusing the test with these work tools.

Existing documentation about testing is numerous, generally with technical methods of items elaboration and processing [1, 2, 3]. With regard to these, in this work, we are not going to value the interest that this kind of objective tests have in the different teaching sectors. There are defenders and detractors who expound their position in a clear and concise form [4]. The only one thing we intend is to give a specification of an open and enlarged model for item edition and management, making the combined use of these tests with LANs and the Internet possible, besides taking us into other fields like the utilisation of on-line adaptive methods to permit them to fit the student profile under examination.

2. BACKGROUND

The tests common denominator is that the student must select an answer between a set of alternatives before performing a mental task sometimes accompanied of the physical manipulation for the response. The relationship between the process of construction of a test and its validity is complex, and precisely because of this complexity it is necessary to establish some mechanisms or rules for their creation. In brief, we can say that, for the creation of an effective test item, we must keep in mind what to check, that is to say, what the variable to determine with the item is. It is no easy task to obtain the maximum relation between the item and the evaluated goal, the ratio that influences directly in the validity and reliability of the outputs. Its determination requires not only mastery of the subject matter and the expression, but also a great knowledge of the psychological aspects of the measured capacity.

The test items can have diverse formats such as true/false, multiple true/false, double answer, limited answer, multiple choice questions, and so on. The latter are the aim of the present model, that is to say, the items in which there are possible multiple answers defined and the user should select the one or the ones he/she considers to be correct. In this respect there exists an extensive bibliography in which the steps to follow for the creation of these kinds of items are specified, besides forms of rules to make the content structure and the canons so the student assessment is complete and does not decrease by a poor or bad definition [5].

A system for items editing has been developed in Díaz Villalta et al. [6] following the methodology established in Neira et al. [1] which serves as the foundation for the model here presented.

3. ITEMS EDITING AND MANAGEMENT

Next, we will specify the bases of a model to make it possible to carry out all the processes that take part in test creation and utilisation by computer. First we have the application which will offer the user all the necessary tools to create and qualify each item that will make up the tests. Once the items have been edited, there arises the need to create the data structures in order to classify and associate these items with clusters of concepts and knowledge areas. To save and support this items framework, different databases are defined. They will be used for items search and selection as well as the elaboration of reports and statistical studies. Once we have reached this point, we can proceed to create the true test, that is to say, the set of items that make up the exercise. We must also give the students the necessary tools for their access to its execution either through a LAN or through the Internet.

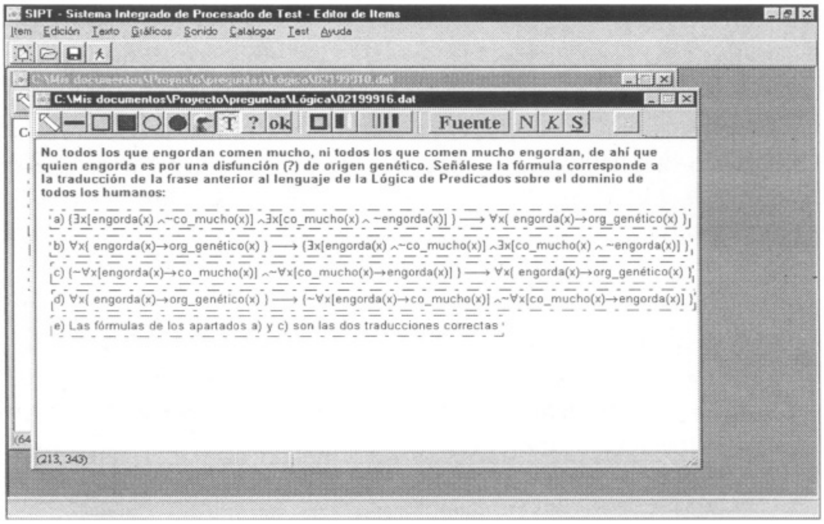
3.1 The items editor

The goal here is to have a tool capable of including the items creation methods and rules for their easy utilisation by the teacher [1, 2]. For the application of these methods it was decided to avoid the items editors which contain the text format of the head, the number of possible solutions and the graphics elements prefixed [7], because we believe that these structures limit the educator when the time comes to embody his/her ideas on a determined item.

Therefore, the application will provide all the necessary tools on the screen so that the teacher can develop and "draw" the elements of the item at will. For this reason, the main window has been divided into six superimposed levels. By means of these, by freely inserting a text in any format, graphics or photo files and drawing elements as lines, rectangles or ellipses it is possible to observe important aspects of items or simply to make it more "attractive". The last level allows us to specify the possible solutions as well as the item correct responses. Finally we could also associate the item with multimedia elements, such as sound, thus enlarging the range of matters the test could be applied to. (figure 1).

3.2 Databases and classification

Once the items have been created, they must be classified inside different structures to facilitate subsequent relationships between the items and the related topics or concepts. To do this, a "catalogue" has been defined. It consists of relating all the topics and concepts which concern a subject matter in a hierarchical structure by way of a tree (figure 2). By means of this item structuring it is possible to have the items classified according to the issues they are related. It is feasible too that an item is associated with several topics within the same classification because to reach the correct answer of some questions it is necessary to apply knowledge about different subjects.


SIPT - Sistema Integrado de Procesado de Test - Editor de Items
Item Edición Texto Gráficos Sonido Catalogar Test Ayuda
C:\Mis documentos\Proyecto\preguntas\Lógica\02199916.dat
Fuente
N K S
(213, 343)

No todos los que engordan comen mucho, ni todos los que comen mucho engordan, de ahí que quien engorda es por una disfunción (?) de origen genético. Señálese la fórmula corresponde a la traducción de la frase anterior al lenguaje de la Lógica de Predicados sobre el dominio de todos los humanos:

a) {∃x[engorda(x) ∧~co_mucho(x)] ∧∃x[co_mucho(x) ∧ ~engorda(x)] } ⟶ ∀x{ engorda(x)→org_genético(x) }

b) ∀x{ engorda(x)→org_genético(x) } ⟶ {∃x[engorda(x) ∧~co_mucho(x)] ∧∃x[co_mucho(x) ∧ ~engorda(x)] }

c) {~∀x[engorda(x)→co_mucho(x)] ∧~∀x[co_mucho(x)→engorda(x)] } ⟶ ∀x{ engorda(x)→org_genético(x) }

d) ∀x{ engorda(x)→org_genético(x) } ⟶ {~∀x[engorda(x)→co_mucho(x)] ∧~∀x[co_mucho(x)→engorda(x)] }

e) Las fórmulas de los apartados a) y c) son las dos traducciones correctas

Figure 1. Overview of the Editor

A second use of this items cataloguing appears when a prefixed test are created. The model allows three methods for item selection. On the one hand, the teacher can select the questions directly, in an analogous way to the classic pencil and paper test. To put it simply, the subject to be examined must be chosen and all the classified items with a related topic will be shown and then those which make up the test must be marked. Another item selection method consists in identifying the topics which make the test, and the system, randomly, will select the items among those available. And finally, the third method is to indicate a catalogue and let the system extract all the items.

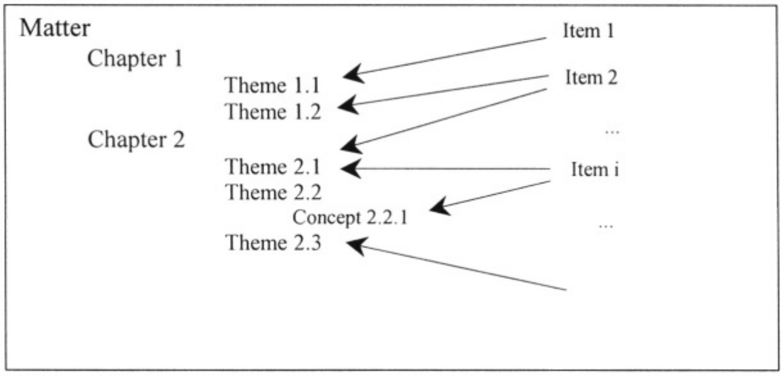


Figure 2. Items hierarchical organisation

3.3 Adaptive selection of items

An adaptive test follows an administration process of test in such a way that the items are presented to the student as a result of his/her activity and his/her previous items answers. An adaptive test evaluates with the same accuracy all the students independently of their *skill* levels, against a conventional test that has its greatest accuracy in the nearest level to the level of average difficulty of the test. To carry out this task it is only necessary to be able to provide at all times items with adequate grades of difficulty and discrimination to the skill level of student. The evaluation produced by an adaptive test is of higher quality and with fewer items than produced by a conventional test; this improvement is reflected in a higher grading accuracy on all the levels of the student's feature that is been analysed; that is to say, there is an improvement of psychometric characteristics of the tests: reliability and validity [8].

There are a lot of research works about adaptive test by computer [9, 10]. The main difficulty to carry out these tests is the choosing of items that must be realised in order to make the test as efficient as possible. As an approach to matter, we have chosen to implement Owen's methodological model [11], because it is the base of great majority of the present commercial implementations [12].

Owen's adaptive test model uses various parameters to characterise the items. These parameters are:

Item difficulty: it values the item conceptual difficulty

Item discrimination power: it values the knowledge of matter if the item is correctly answered.

Item guessing factor: it values the possibility to answer correctly if the student has not got any knowledge of the matter.

The parameters to the final valuation of the student are based on a normal distribution function and a density function characterised respectively by parameters ϕ and Φ.

```
Begin
• to choose a theme to evaluate;
• to introduce the parameters initial values:
      initial average: is an estimation of the student's knowledge about
      the theme;
      initial variance: is an estimation of the confidence interval to
      approach the average;
      delta: the chosen item will verify that the absolute value of
      difference between the chosen item difficulty, $d_n$, and the average,
      $M_{n-1}$, must be minor than delta; which is:

                  $\mathrm{Abs}(d_n - M_{n-1}) < delta$;

      finishing_parameter: if the current variance is equal or minor to
      finish the parameter, the process is stopped.
While variance > finishing_parameter do
• to choose the item with maximum information, which verifies Abs(dn -
  Mn-1) < delta, and also it has not been chosen previously in the course
  of the test;
• to assign the values to parameters formulas:
      g:= item guessing factor    (1/ number possible of item answers)
      d:= item difficulty
      $M_0$:= item discrimination power
      $V_0$:= current variance;
• to check the answer of student to the current item;
• if the answer was correct then to calculate the new average and the new
                                 variance according to formulas in figure
                                 3-a
                            else according to formulas in figure 3-b
end_if
end_while
to calculate the last mark
end.
```

a)

$$E(\theta/1) = M_0 + \frac{(1-g)V_0\sqrt{\frac{1}{p^2}+V_0}\,\phi(D)}{g+(1-g)\Phi(-D)}$$

$$Var(\theta/1) = V_0\left\{1-(1-g)\frac{1}{1+\frac{1}{p^2}\frac{1}{V_0}}\phi(D)\frac{\left[\frac{(1-g)\phi(D)}{A}-D\right]}{A}\right\}$$

b)

$$E(\theta/0) = M_0 - \frac{V_0\sqrt{\frac{1}{p^2}+V_0}\,\phi(D)}{\Phi(D)}$$

$$Var(\theta/0) = V_0\left\{1-\frac{1}{1+\frac{1}{p^2}\frac{1}{V_0}}\phi(D)\frac{\left[\frac{\phi(D)}{\Phi(D)}+D\right]}{\Phi(D)}\right\}$$

Figure 3. Main formulas of Owen's method

3.4 Client/Server application of items

Once the items classification subsystem and the trial producer/manager subsystem have been built, the next task is to build a process that allows the communication with students by means of networking (intranet, Internet, etc.). An option based on an architecture client/server and on the Internet proper technologies is presented below.

Two tools carry out the task. An administrator tool works as *Web Server* and the other tool allows us to access all resources offered by the administrator tool; that is to say, it works as a *Web Client* (figure 4).

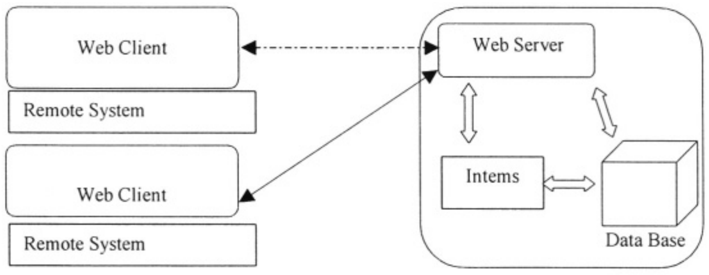
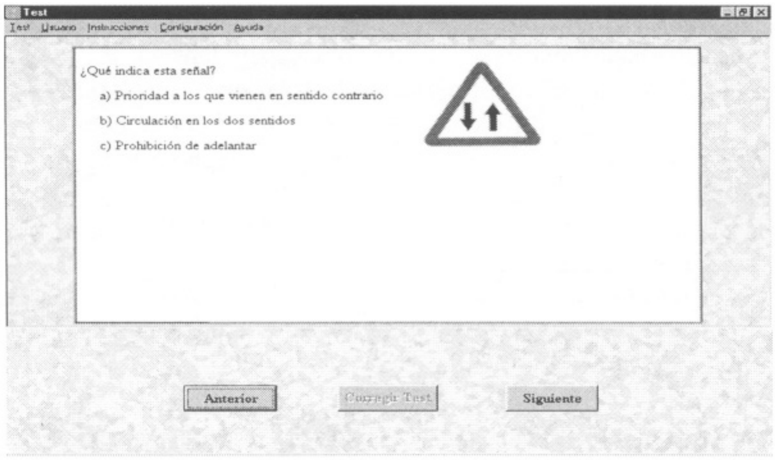


Figure 4. Client/Server Architecture

The server tool allows us to exchange the information between the *DataBase* and the last user. It administers all the requests received and it stores in the *DataBase* all the dates accumulated by the last users: answers, marks, etc.

The *Client* application (figure 5) is distributed among students and it allows them access to the test generated by the *Server.* Once the communication *Client-Server* is established and the user's identity is recognised, behind a series of steps the system allows the student to start the adaptive test, according to Owen's model. In the end, the *client* application sends the results to the *DataBase.* After this, the last mark is shown. Also, by means of this tool, each student can have access to all the work they themselves have done.

There is the possibility to carry out a classical non-adaptive test.



¿Qué indica esta señal?

a) Prioridad a los que vienen en sentido contrario

b) Circulación en los dos sentidos

c) Prohibición de adelantar

Figure 5. Overview of the Client

4. TRIALS AND CONCLUSIONS

At present, there are two catalogues with items edited and classified about matters such as logic and knowledge of traffic codes corresponding to official tests to obtain a car driving licence. The logic catalogue includes basic knowledge to assess the *Logic* matter of first course of the *Escuela Universitaria de Ingeniería Técnica Informática* at the University of Oviedo in Gijón (Spain). The pupils use it as a tool to pass the tests. They can work through the university intranet from their work place at school, or from home through the Internet.

Initially, the system was tried out with connections from a work place at school through the intranet of the University of Oviedo; it was also tried with Internet connections from far away places in Spain and from foreign places. The system always responded in the expected way .

A system has been built up with various advantages both for researchers and students alike.

Among the advantages for instructors the following can be pointed out:

- By means of a system editor it is possible to edit items in a way which is easy and comfortable, practically without limitations of forms, answer number, colours, graphic, sound, etc. in accordance with present methodologies.
- The cataloguing of items is open and flexible, allowing the test making to be directly accessible to students training.
- The system allows working in network, with direct communication of the students with the Database.

- The system allows the provision, in a structured way, of all the information about the tests and results of each student.

Among the advantages to the students the following can be pointed out:

- They can access from different work places: at School, at home, and from anywhere with connection to the Internet; so as to check their knowledge during training.
- The system environment is friendly, it motivates the students to train and it is helpful when studying.
- It is possible to work on a network to interchange the opinions about the results of trials.

5. FUTURE DEVELOPMENTS

After the model has been presented and its installation initially tested, it is necessary to evaluate its benefits. Because of this other comparative studies will be carried out with other adaptive methods. Also, an automated management will be carried out to emit reports both for students as for instructor; it could be used to obtain information to realise processes of machine learning, whose results will help to classify the student. It is planned to incorporate more multimedia elements such as video, and in this way, the area of application of the objective tests will be extended.

REFERENCES

1. Neira T.R.(Coordinador). *La Evaluatión en el Aula.* Ediciones Novel, Oviedo, 2000. Spain.
2. Neira, T.R.; Albuerne, F.; Alvarez, L.; Cadrecha, M.A.; Hernández, J.; Luengo, M.A.; Ordóñez, J.J.; Soler, E. *Instrumentos de Evaluatión de Aprendizajes.* I.C.E. Aula Abierna mon. 22. 1993. Oviedo. Spain.
3. Osterlind, S.j. *Constructing Test Items.* Kluwer Academic Publishers. Boston. 1989.
4. Soler, E.; Álvarez, L.; García, A.; Hernández, J.; Ordóñez, J.; Albuerne, F.; Cadrecha, M.A.; *Teoría y Práctica del Proceso de Enseñanza-Aprendizaje.* Narcea, 1999, Madrid.
5. Vaquero, A.; *La Enseñaza Asistida por Computadora. Métodos Informáticos Aplicados a la Psicología.* Eds. Algarabel, S.; Sanmartín, J. Pirámide 1990. Spain
6. Díaz Villalta, J.A.; Neira A.; Alguero A.S.; L.Brugos, J.A.; García V. ; Álvarez I.; Soler E.; *Sitema Integrado de Procesado de Test.* I.C.E. Aula Abierta n° 74, Oviedo 1999. Spain
7. Millán, E.; Trella, M; Pérez de la Cruz, J.L.; Conejo, R.; *Uso de Redes Bayesianas en Test Adaptativos Computerizados.* Congreso Nacional de Informática Educativa. Puertollano (Ciudad Real), 1999. Spain.
8. Neira, A.; Alguero, A.S.; Brugos, J.A.L.; García V.; *Approach to Intelligent Adaptive Testing and Optimized Fuzzy Logic Model.* In Computers and Education in the 21th

Century. Edts. Ortega, M; Bravo, J.; Kluwer Academic Publishers, Dordrecht, Netherlands 2000.

9. Sánchez, Pedro J.; Martínez, Luis. Dolores Muñoz, Mª.; *Un Sistema de Generación y Evaluciación de Exámenes Basado en Java* Congreso Nacional de Informática Educativa. Puertollano, (Ciudad Real) 1999. Spain

10. López, J.A.; Ato, M.; Sánchez, J.; Velandrino, A.; *Tes y Diagnóstico Psicológico por Computador.* In Métodos Informáticos Aplicados a la Psicología, Edts. Algarabel, S.; Sanmartín, J.. Pirámide, 1999. Spain

11. Owen, R.J.; *A Bayesian Sequential Procedure for Quantal Response in the Context of Adaptive Mental Testing.* Journal of the American Statistical Association, June 1975. p. 351-356.

12. Huang S.X.; *On Contenet-Balanced Adaptive Testing.* CALISCE'96, Donostia 1996. Spain.

Symbolic Calculus Training by Means of *Math Trainer*

Fernando Díez, Roberto Moriyón
Departamento de Ingeniería Informática. Universidad Autónoma de Madrid – 28049 Madrid, Spain
Fernando. Diez@ii.uam.es, Roberto.Moriyon@uam.es

Key words: computer assisted learning, symbolic computations, programming by example.

Abstract: This paper describes *MathTrainer,* a system that guides the student while learning mathematical concepts that involve symbolic manipulation, such as the resolution of Ordinary Differential Equations. *MathTrainer* shows students the steps that are required to solve exercises with symbolic calculations. The exercises can be posed by the student or by the system. *MathTrainer* identifies the type of exercise by means of a pattern matching mechanism and then it successively shows the tasks to be accomplished in order to solve the exercise. *MathTrainer* has been integrated in the *MathEdu* teaching environment, which includes an authoring tool for the design and interactive resolution of mathematics exercises. Thanks to this integration, *MathEdu* allows the students to learn first how to solve problems of different types; after this, they can practise and get feedback about their actions.

1. INTRODUCTION

During the last years new tools have been developed that simplify the design of generic user interfaces [1], using *Programming by example* techniques [2]. In principle, these techniques can be used in an authoring tool in order to make the work of the teacher who designs the contents of the course more intuitive and simpler to accomplish. There are two means that allow the designer of a course to work using programming by example: on the one hand, the tool can allow the designer to work in the same environment the student will work when following the course. This kind of environment facilitates the teacher's discovery of the difficulties the student can have when working by himself, as well as the enhancements that can be incorporated to the course under development. On the other hand, the mechanism of generalisation allows the teacher to work just on a specific example or problem and later on to extend his/her work to a whole set of situations that include the original one as a special case. The development of a module with this feature for an authoring tool is very complex, but it is a task that has to be accomplished just once, and it can be done by experts without big limitations about the amount of time to be spent on it. Once this work has been done, it is reused constantly by the different course designers in each course they develop. Moreover, in this way the course designers do not need any special knowledge about the language or technology that is used behind the scenery, so most teachers can use the tool without any special training.

In this paper we describe how we have enhanced a tool for the interactive resolution of problems of Mathematics, *MathEdu* [3], using the techniques described in the previous paragraph, with a new module, *MathTrainer*, that guides students through the steps required to solve exercises which can be specified dynamically. The exercises can be posed by the student or by the system. *MathTrainer* identifies the type of exercise by means of a pattern matching mechanism and then it successively shows the tasks that have to be accomplished in order to solve the exercise. Thanks to the integration of *MathTrainer, MathEdu* allows the student to first learn how to solve problems of different types; after this, s/he can practice and get feedback about his/her actions. In case the student repeatedly makes some mistake while s/he is practising, *MathTrainer* takes control of the session and shows him/her how to finish the resolution of the problem s/he is trying to solve. This is related to the *MathCAL* approach, [4]. The enhanced *MathEdu* system can be used to develop complete interactive courses about mathematical subjects that involve symbolic manipulation, such as the resolution of Ordinary Differential Equations.

The paper is divided into the following sections: first, the main features of *MathTrainer* are described. After this we show the mechanism used for the representation of the symbolic aspects of mathematical knowledge, and the way this mechanism is used by *MathTrainer*. Finally, we present some conclusions and references.

2. THE USE OF *MATHTRAINER* BY THE STUDENT

MathTrainer is used by a student who follows an interactive course. A *MathEdu* course is a sequence of chapters that include theory, examples, and problems. Examples are instances of problem types with specific values for the formulae that appear in their statements. While working, students first read the theory and the examples, and then they can either ask for more examples, which are dynamically generated, or solve some problems in a guided setting. The sequence of exercises is not predefined: the student selects the subject, and the type of exercise is chosen randomly by the system.

From the point of view of the student, *MathTrainer* is just a program that allows him/her to sequentially visualise a presentation of the steps required to solve an example. At the beginning of an explanation the student can choose the problem to be explained by typing some of the formulae that appear in it. For example, when seeing how to solve a linear differential equation, and after having seen a simple example, such as the resolution of the equation $y'+y=1$, the system will explain that any function of x and y that is linear in y can appear instead of y, and any function of x can also appear instead of 1. The student can then ask the system to show him how the solutions of equations like $y'+y/x=x^2$ can be computed, and *MathTrainer* will then give him/her a step by step explanation of the resolution of this problem. In case the example proposed by the student is not appropriate, the system will tell him/her that, and it will also let him/her know whether his/her example can be already solved according to what he/she has already studied with *MathEdu.*

At any moment the student can also ask for the origin of some formula that appears in the explanation. This is done by selecting the desired mathematical expression and using a tool for this purpose. The system then will show him/her the formula that has been used in order to compute the expression, as well as the previous expressions that are involved in the computation.

Another way in which *MathTrainer* is integrated with the previous *MathEdu* module for interactive problem resolution by the student is by giving him/her explanations about how to go on in case of a repeated mistake while solving a problem, or in case he/she just asks for this help. In the original version *of MathEdu,* this kind of help could be provided only in case the designer of the course had foreseen its necessity, and in this case, the designer had to specify the whole explanation *by hand.* This limitation of *MathEdu* was especially important, since the same situation appeared at every step of the student's work, so the amount of work involved if the designer wanted to give support for the student to get additional explanations at any moment of his/her work was tremendous.

3. THE USE OF *MATHTRAINER* BY THE DESIGNER

The design process of courses by means of the enhanced *MathEdu* environment makes the presence of *MathTrainer* completely transparent to the teacher who is in charge of it. Thanks to the use of *Programming by Example* techniques, the designer only has to concentrate on the problem resolution process as the student will face it and the sequencing between the different kinds of problems instead of thinking about the internal aspects of the resolution management algorithm. In this section we shall give a brief description of this process.

The design of a course is done through *Mathematica®* notebooks [5], which are interactive documents composed by cells that allow the insertion of text, mathematical formulae, buttons and graphics, as well as palettes with buttons and tools for the accomplishment of some special tasks. The theoretical contents of each chapter are introduced using the standard procedures provided by *Mathematica®* notebooks. Problem types and examples are introduced by typing their statement, and generalising them by associating parts of the mathematical expressions that appear in the statement to generic metavariables which include acceptance conditions and random generating functions of the expressions involved in the statement. The designer then specifies the resolution process for the problem and the guidance to be offered to the student while solving it by explicitly indicating a sequence of dialogues that allow the student to determine corresponding data at each step during the process. The value of each new data that the student has to determine is specified in terms of symbolic operations applied to the previous ones and evaluated at run time. These dialogues, together with some minor variants of them, and subproblems to be solved form a sequence of actions that the student and the system perform during the resolution process.

The formulae that indicate the values that the student has to introduce at each step play a role similar to that of formulae in spreadsheets. The role played by cells in spreadsheets is played by mathematical expressions in *MathEdu*. The main difference between the two cases, besides the power of symbolic calculation obtained by the use of *Mathematica®*, lies in the fact that *MathEdu* specifies an order in which the corresponding mathematical expressions are shown to the student.

The design process can be seen as if the designer was doing by hand the actions that both the system and the student are supposed to do at resolution time, and the design engine recorded these actions. When *MathTrainer* is showing the student how to solve a problem, either generated by *MathTrainer* itself or posed by the student, it just reproduces the actions specified by the designer and makes itself the computations that are involved. On the other hand, when the student is solving a problem, *MathEdu* just executes the actions specified by the designer, corresponding to the system, and it checks that the expressions introduced by the student through the dialogues are all right according to the corresponding formulae. Every expression introduced by the student in an action is contrasted either with a pattern for such an expression that has been specified by the designer or with an expression obtained from the metavariables through symbolic operations. This validation process uses the powerful pattern matching mechanisms of Mathematica, that allow the specification of arbitrarily complex structural patterns for mathematical expressions. For example, a relatively simple pattern in Mathematica can exactly match rational functions.

MathEdu also uses the power of symbolic computation of *Mathematica*® to allow the designer to define strategies and cases for each type of problem depending on the specific data associated to the problem. A strategy includes patterns for the metavariables of the problem. These patterns are used to check when the strategy can be applied. Strategies also include resolution actions representing the steps that have to be accomplished in order to solve the exercises.

Different strategies for the same type of problem differ in the actions that have to be accomplished in each case, while different cases for the same strategy differ by the pattern that characterises the values of the initial metavariables that are present in the statement. For example, linear ordinary differential equations correspond to the pattern

```
    a_var' + f_function[b_var] a_var = g_function[b_var], (1)
```

which is specified by the designer.

Both when showing the students how to solve a problem and when checking how they solve it, the first thing *MathEdu* does once the specific statement has been specified is to determine the possible strategies and cases that correspond to the problem. This is done by means of a pattern matching process between the formulae that appear in the statement and the patterns that correspond to each strategy. For example, if the problem consists of solving the differential equation

$$y' + y/x = x^2,$$

then the metavariables a and b in expression 1 are associated to the variables *y* and *x* respectively, and the functions f and g are associated to $1/x$ and x^2 respectively, hence the strategy for resolution of linear differential equations is accepted, which has only one case. The sequence of actions that have to be accomplished in this case is the following one:

- Solve the following subproblem: solve the corresponding homogeneous linear equation.
- Substitute the multiplicative constant in a generic solution to the homogeneous equation by a generic function C of x, thus obtaining a new function that will be equal to y.
- Substitute the resulting expression into the differential equation, and solve the resulting equation for C.
- Substitute the value obtained for C into the expression for y. This is the solution to the problem.

These actions are specified interactively by the designer. Each action will give rise to a presentation in case the system is showing the student how to solve the problem, but if it is checking how the student solves it, a dialogue for each step will be shown, and then it will check the answers given by the student to the questions asked in the dialogue.

4. CONCLUSIONS

In this paper we have shown the main features of a highly interactive system to train students in solving problems of Mathematics. *MathTrainer* shows the students how to solve problems generated by the system and others that are posed by the students themselves. The specification of the way the system has to show the students how to solve each type of problem and how it has to check the students' resolution of each problem is done through the use of a variant of the *Programming by Example* paradigm.

MathTrainer incorporates new functionality that was not available in previous tools. This functionality facilitates the definition of courses on subjects that involve symbolic computation allowing the student to learn the resolution methods for exercises related to these subjects faster. The main goal is to make the student feel the system as a collaborator in his/her learning process.

At this point there is a first prototype of *MathTrainer* that is partially integrated with *MathEdu*. The first version of the enhanced *MathEdu* environment will be ready in some months, and the validation of the system will be done through the development of a course on Ordinary Differential Equations.

Finally, among the future work we have plans to develop mechanisms that allow the system to automatically detect the existence of priorities between different parts of the course, so that if a student wants to learn some special subject, an optimal path is created automatically that will show him all the necessary prerequisites before teaching him the desired subject.

ACKNOWLEDGEMENTS

This paper has been sponsored by the Spanish Interdepartmental Commission of Science and Technology (CICYT) project numbers TEL1999-0181 and TEL97-0306.

REFERENCES

1. Myers, B., McDaniel, R., and Wolber, D. (2000). *Programming by example: intelligence in demonstrational interfaces.* Communications of the ACM, Vol. 43, No. 3, pp. 82 – 89.
2. Cypher, A. (1993). *Watch what 1 do. Programming by Demonstration.* Cambridge, MA: The MIT Press.
3. Díez, F.; Moriyón, R. (1999). *Doing Mathematics with MathEdu*, Proceedings of the IXth Conference of Mathematics/Science Education & Technology. AACE. San Antonio (Texas).
4. Mei-Chuen, J.; Juang, J.; Sun, P. (1999). *An Internet-Based CAL Software for Solving Trigonometric Problems*, Proceedings of the IXth Conference of Mathematics/Science Education & Technology. AACE. San Antonio (Texas).
5. Wolfram, S. (1999): *The Mathematica Book*, 4th Edition. Cambridge University Press.

EJS: An Authoring Tool to Develop Java Applications

F. Esquembre, J.M. Zamarro
University of Murcia. Campus de Espinardo, 30071 Murcia. Spain.
E-mail: fem@um.es, jmz@um.es

Key words: authoring tool, Java

Abstract: The use of computers for teaching, and most specifically in learning Physics, may be an element that will improve and drastically modify the traditional teaching process. Easy Java Simulations, EJS, is an authoring tool that allows people that are not programming expert and unknowing the Java language, to develop Java applications. EJS structure is based on the idea that most computer simulations of scientific phenomena can be described in terms of the so-called Model-Control-View paradigm.

1. INTRODUCTION

Computer simulations could be a very efficient element in the learning process [1]. The lack of tools that allow teachers that are not expert in computers to develop their own applications is one of the reasons why computer simulations are not so used in the teaching process. EJS is a product for the teacher/author that allows the generation of interactive graphic environments with high realism, facilitating the use of powerful programming structures with the simple use of the mouse without needing to write a code line. The applications produced with this tool are Java products, so they are independent, multi-platform, they can be visualised using any Web browser (and therefore distributed through the Internet), read data across the net, communicate with other applets and be controlled using scripts from within html pages. This last characteristic is very important to develop didactic units to promote discovery learning [2].

EJS structure is based on the idea, previously implemented by the author in UNIX environments [3] and by other applications [4], that simulations are basically formed by three components: Model, View and Control.

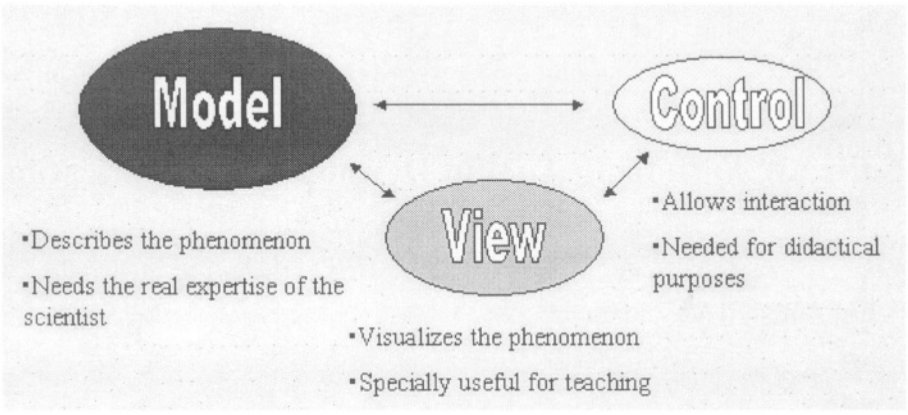

Model
Control
View
•Describes the phenomenon
•Needs the real expertise of the scientist
•Allows interaction
•Needed for didactical purposes
•Visualizes the phenomenon
•Specially useful for teaching

Figure 1 Parts of a simulation

It is taken for granted that the user knows how to translate their model by means of some subgroup of sentences of C language, defining actions linked to programming structures they can create their graphic interface, allowing time running interaction with the model. EJS hides most of the Java aspects. EJS is Java written, so it allows the user to concentrate her efforts in the simulation model. The most appealing fact for people that begin with this kind of environments is the View. It allows programming sophisticated graphic simulations without writing a code line.

2. DESCRIPTION OF THE SIMULATION ENVIRONMENT

When running EJS a window with three different areas appears. On the upper side we see a menu bar with the entries File, Run, Font and Help, and a toolbar with some icons, which are standard in most programs. They allow us to open, to save, to generate applications and to run them, among other possibilities. Just below the toolbar we see a panel with four tabs labelled Model, Control, View and Connections. When the Model tab is selected (which makes it grey, although it is red when unselected), we can see again a panel with five tabs. This means that the Model part of our simulation can consist of up to five subparts. The other tabs, the Control, View and Connections tabs do not have the same structure, but a specific one adapted to their nature, as we will see. At the bottom there is a panel where the messages that generate EJS are shown as our instructions are processed.

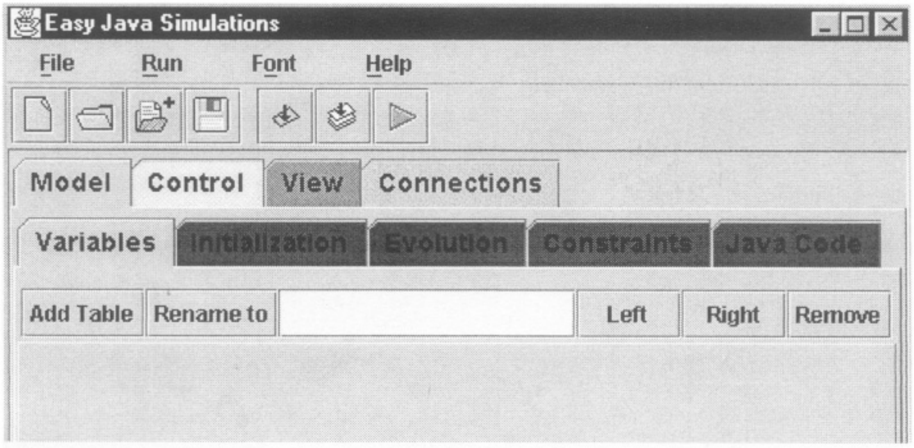
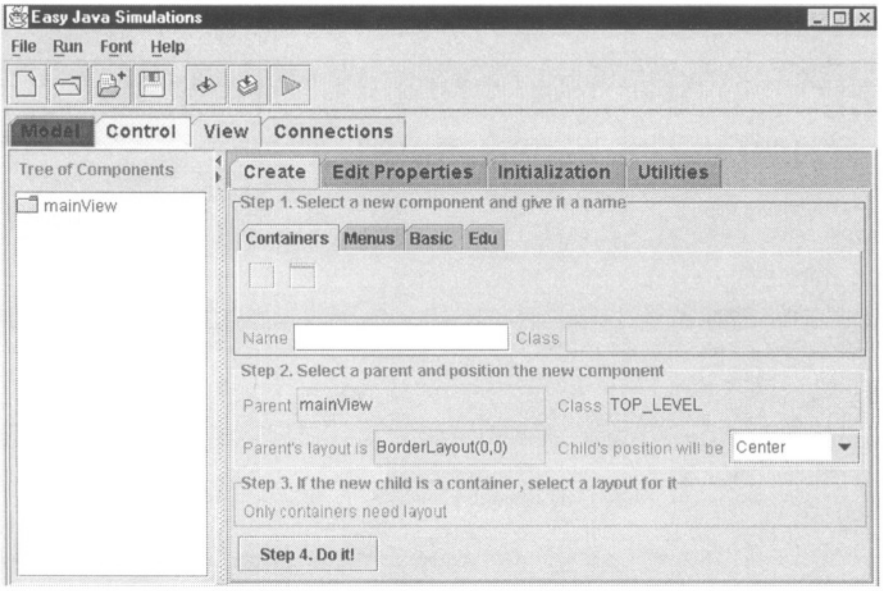


Figure 2. EJS Initial Interface

Figure 2 shows the parts to implement the phenomenon we want to simulate. We create the model of a phenomenon when we identify the relevant magnitudes, set their values at a given moment and establish the laws that govern how these magnitudes change. We use the word magnitude to refer both to state variables (values that describe the phenomenon) and to parameters (values used by the governing laws). When we write our simulation, we will refer to magnitudes as variables. The system can evolve autonomously from the current state to a new one simulating the passage of time (which, by the way, may or may not be one of our variables). We call the equations ruling this transition evolution equations, which can be written as a system of one or more equations. Simulating the evolution of a model in time consists in computing the new values from the current state of the model, taking these as the new state of the model, and continuously repeating this process for as long as the simulation runs.

A view is created by including graphical components within a window. The View gives us the opportunity to create a sophisticated graphic interface with practically the only use of the mouse. To facilitate this process the View has been divided in some parts, first the Creation of objects shows us a main window that will be used to implement our interface, components are grouped by their features. Once a component has been created we can edit their characteristics and modify them at will. Other panel lets us write the code we want to be run when the interface is initialised. Finally, Utilities give us the opportunity to remove the component or modify the characteristics generated as it was created.


Easy Java Simulations
File Run Font Help
Model Control View Connections
Tree of Components
mainView
Create Edit Properties Initialization Utilities
Containers Menus Basic Edu
Step 4. Do it!

Step 1. Select a new component and give it a name
Name Class

Step 2. Select a parent and position the new component
Parent mainView Class TOP_LEVEL
Parent's layout is BorderLayout(0,0) Child's position will be Center

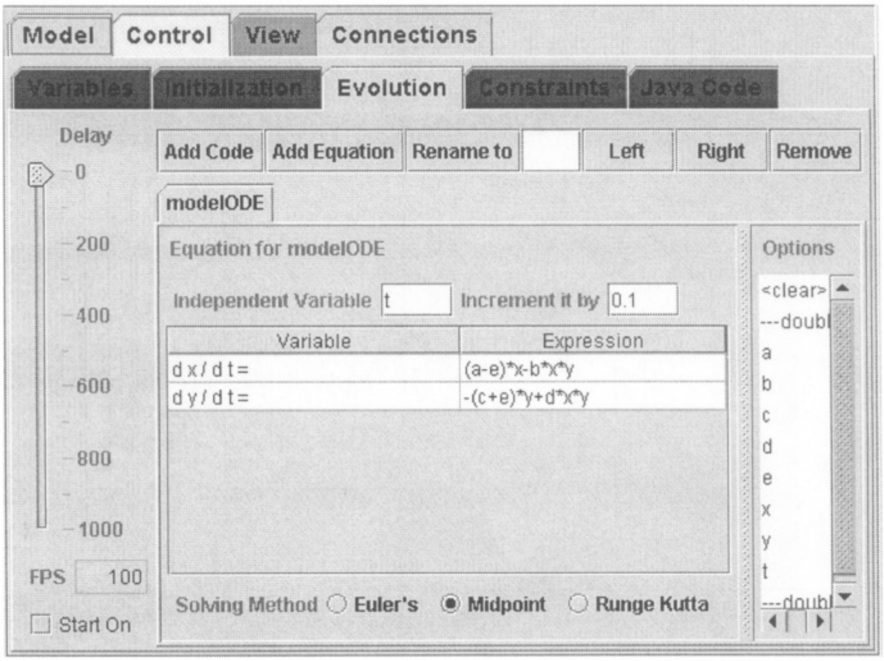
Step 3. If the new child is a container, select a layout for it
Only containers need layout

Figure 3. Panels associated to the creation of components.

An important fact of the simulation is their interactivity, that is, to be able to change the model variables from the graphic interface and also some characteristics of the their components, to do this EJS has Connections that allow us to link characteristics of the graphic interface with the model variables or with actions previously defined in Control.

3. AN EXAMPLE: PREDATOR-PREY SYSTEM

We will give an example that will show us one of the features of the model, the resolution of differential equations writing the mathematical expressions as it is shown in figure 4.



Equation for modelODE

Independent Variable t Increment it by 0.1

Variable	Expression
d x / d t=	(a-e)*x-b*x*y
d y / d t=	-(c+e)*y+d*x*y

Solving Method Euler's [✓ Midpoint] Runge Kutta

FPS 100

Start On

Figure 4. Add Equation at Evolution in Model allows writing differential equation.

Let us guess x(t) and y(t) represent the prey and predator population at instant t, without predators the prey population will evolve proportionally to its value, $x' = ax$, being $a>0$, the number of contacts predator prey could be written as bxy, so we can write the variation of the prey population as $x' = ax - bxy$. In a similar way the number of predators will grow proportionally to their population y and to food x, and will decrease with dying, $y' = -cy + dxy$. If we introduce the effect of hunting in the same rate for both populations, their evolution could be expressed by the following equations:

$$\begin{cases} x' = (a-e)x - bxy \\ y' = -(c+e)y + dxy \end{cases}$$

These equations could be introduced in this format in the Model selecting Add Equation as it is shown in figure 4. Using View we can create a graphic interface similar to the interface shown in figure 5.

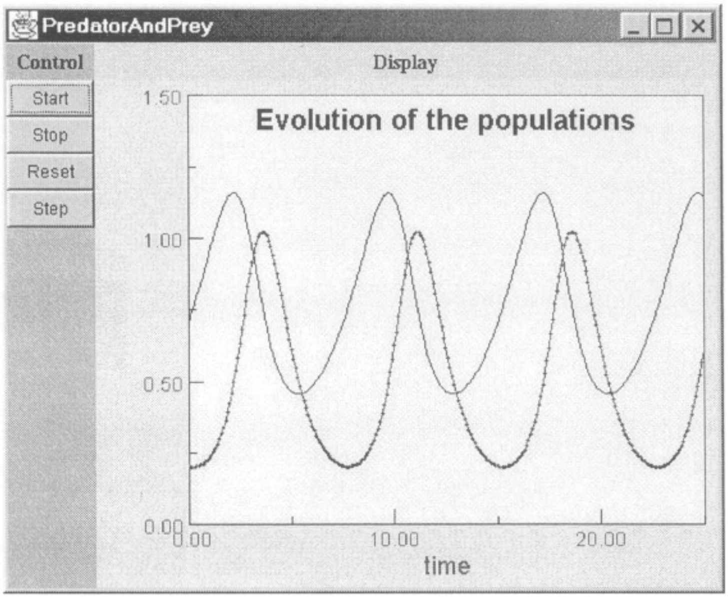


Figure 5. A graphic interface for the simulation of the predator-prey population evolution

EJS is a product for free distribution. At http://fem.um.es/Ejs/index.htm you can find information about this tool as well as download facilities with some examples, the one we have shown in this paper included.

ACKNOWLEDGEMENTS

The authors want to express their acknowledgment to the Ministerio de Educación y Cultura, project 1FD97-1803, and to the Comunidad Autónoma de la Región de Murcia for their economic support for this work.

REFERENCES

1. de Jong, T. "Learning and Instruction with Computer Simulations". Education & Computing, 6, 217-229, (1991).
2. Njoo, M.K.H. Exploratory learning with a computer simulation: learning process and instructional support. Thesis, Technische Universiteit Eindhoven. (1994).
3. Zamarro, J.M., Martín, E., Esquembre, F. Authoring System for Flexible Developments and Some Applications. IFIP open Conference. Informatics and Changes in Learning. Gmunden, Austria, junio (1993).
4. Duarte, V. http://phoenix.sce.fct.unl.pt/modellus/

ADAM CASE
Using upper CASE tools in Software Engineering Laboratory[✝]

F.J. García, M.N. Moreno, A.Mª Moreno, G. González, B. Curto y F. J. Blanco
Departamento de Informática y Automática - Universidad de Salamanca

{fgarcia, mmg, amoreno, gyermo}@gugu.usal.es; {bcurto, jblanco}@abedul.usal.es

Key words: CASE tool, structured paradigm, object-oriented paradigm, DFD, class diagram, UML, Software Engineering laboratory

Abstract: In general, CASE tools are a fundamental support to improve and automate the software development process in every stage of the project life cycle. From this point of view, it seems necessary that the future software engineers should be trained in the appropriate techniques to build software applications, but they must also take the opportunity to apply these techniques with the help of automated tools. However, these tools are not always available for teaching purposes, because of two reasons: the high cost of the tools licences and the enormous diversity in this area. In this article, ADAM CASE is presented. It is an upper CASE tool developed at the Computer Science and Automatics Department of the University of Salamanca. The main aim of the work is to solve the existing lack of CASE tools in Software Engineering laboratory course within technical studies of Computer Science in this University.

1. INTRODUCTION

Computers have become an essential work-tool in practically every social or labour sector. Adequate software exists in every area, which makes it easier to automate the activities to be done.

The software development industry is not free from this situation; it needs tools to improve software engineers' productivity in the whole life cycle of the project. These tools usually receive the generic name of CASE (*Computer Aided/Assisted Software/System Engineering*) tools.

[✝] This work has been partially financed by SA02/00F Junta de Castilla y León project.

This fact is presented at the moment to organise the teaching in the courses with more emphasis in code related stages, where the exhaustive use of IDEs (Integrated Development Environment) is a common practice in their laboratories.

However, when we go up in the abstraction level of the software definition, automated tools poorly support the imparted concepts and process in the theoretical part.

Several arguments could be given to justify the previous assertion, but we can resume them in economic terms: *laboratories with licences of this kind of tools for every machine are so expensive that they are impossible to get.*

This situation is specially serious in courses related to Software Engineering (SE) discipline, where the utilisation of upper CASE tools is very interesting to automatically support the analysis and design stages.

In this work ADAM CASE v1.0 is presented. It was conceived to be usually applied (but not as the only way) in the SE laboratory course in the technical studies of Computer Science (*Ingeniería Técnica en Informática de Sistemas*) at the University of Salamanca (Spain).

The rest of the article is organised as follows: the second section explains how the tool is used in practice within the SE laboratory course in the context of the Salamanca University. Section three is centred on the exposition of the most interesting technical and functional characteristics of ADAM CASE v1.0. The fourth section lists other related works, comparing them with our tool. Finally, section five closes the paper with conclusions and future evolution of this work.

2. PRACTICAL SOFTWARE ENGINEERING TEACHING SUPPORTED BY CASE TOOLS

Despite the initial failure of the first generation of CASE technology in the eighties, mainly due to its limitations and its bad assimilation and establishment, a new generation of automated tools has come up. Many of these tools are not called CASE, trying to escape from the old ghosts. They try to contribute with better services to build up a systematic software development process.

Nevertheless, although the advantages to automate the software development process are obvious, several reports [1,2] are not very optimistic about the massive utilisation of this kind of tools applied into the information systems of organisations. They emphasise the main use of programming integrated environments compared with the tools that support other activities in the software life cycle, where software is thought or manipulated at a higher abstraction level.

In spite of this, from a curricular perspective, an integration of the CASE technology in the curricula of the computer science courses is recommended [3]. We must care for a better adaptation of the new professionals' qualifications to work with these automated tools, looking for a better software quality, and also a better development process when these begin their professional lives.

Specifically, in SE course, imparted in the third year of the technical studies in Computer Science in the University of Salamanca, we try to make the students know and use CASE tools. Obviously, due to time limitations, we have chosen to emphasise the use of upper CASE tools. This kind of tools is centred in analysis and design stages of software development. For most detailed information about CASE tools classification the next references are suggested [4, 5, 6].

In the organisation of the SE laboratory course, CASE technology is not our mainstay. We intend it to be a support tool to apply specification and modelling techniques, both of structured and object-oriented paradigms, studied in the theoretical part of the SE course. In this way, these automated tools should be used as help and validation in the creation of technical documentation of the practical exercises that are proposed to the students in the SE laboratory.

In the first editions of the SE course, the fulfilment of this objective was compromised by the lack of tools to provide to the students. These shortages were caused by the following reasons:

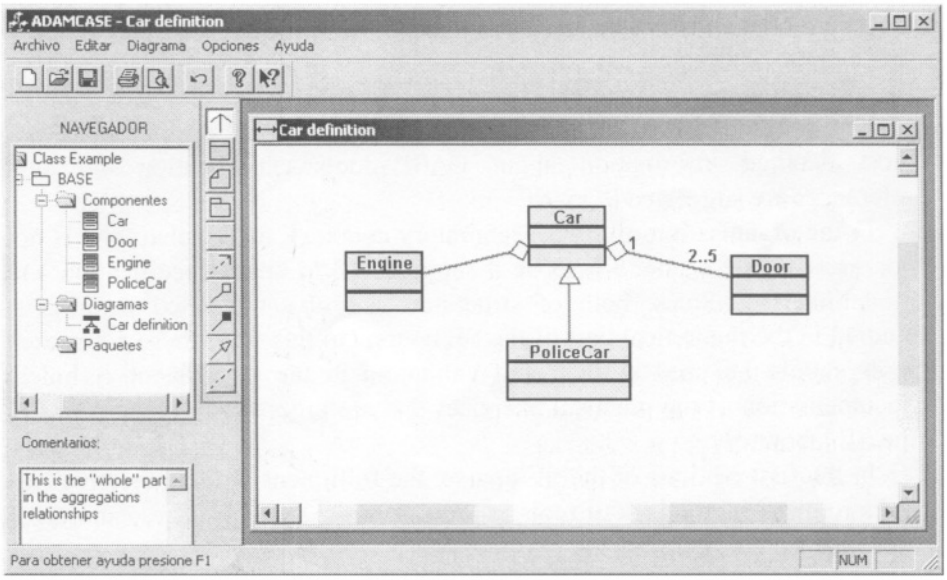
- In the commercial area there were different alternatives, but it was very expensive for a laboratory to buy enough licences of two CASE tools (one for the structured techniques and another for the object-oriented ones), especially when the SE course was not oriented to these tools.
- In the free software area, for the object-oriented paradigm techniques there were many and diverse possibilities, each one with its peculiarities and limitations. However, for the structured paradigm techniques the offer was very poor, some demo versions were available, but they were limited in their facilities.

To face this situation, we decided to recommend some free tools with UML (Unified Modelling Language), like Rational Rose [7] or ArgoUML [8], for the object-oriented part; also we cited some graphical editors with different structured methodologies notation support.

Clearly, this was not the most suitable situation to face up the applied practices in the laboratory in concordance with the proposed teaching goals. For this reason we decided to design and implement a set of CASE tools to apply them both in teaching and in research projects.

As our initial goal, we planned to achieve a first version of an upper CASE tool that supported the two main techniques explained in SE courses, i.e., DFDs (Data Flow Diagrams) and UML class diagrams, both under the same GUI. This should make it easier to accept and help in the formation of its users (students and teachers).

The result of this initial attempt is ADAM CASE v1.0, whose interface is shown below. This tool will be used for teaching in the SE laboratory in academic year 2000-2001.



This is the "whole" part in the aggregations relationships

Para obtener ayuda presione F1

Figure 1. ADAM CASE interface

3. ADAM CASE FEATURES

ADAM CASE v1.0 could be considered as an upper or front-end CASE tool, working under Windows 9x, NT or 2K platforms. It assists developers both in the creation of DFDs, using the Yourdon's notation [9] and in the creation of UML class diagrams [10]. Independent of the diagram type, the tool provides a data dictionary and automatic creation of technical reports.

In this section we shall briefly describe the most outstanding characteristics of ADAM CASE, specifying the technical aspects considered in its development and the main functional facilities.

3.1 Technical issues

This tool has been fully developed under the object-oriented paradigm, using MS Visual C++ 6.0 for its final implementation.

One of the main design restrictions was the integration of DFD and class diagram techniques in a single software application with the same GUI, making the tool management easier.

In the GUI design the next constraints were considered:

– The GUI should be functionally independent: separating the tool interface from its functionality is the way to achieve a portable application to other platforms. The basis of the GUI design is the Model-View-Controller (MVC) pattern, which is widely studied in [11].
– The "look and feel" of the tool interface does not vary too much with other commercial tool interfaces. We have chosen the Rational Rose [7] interface as a model because it is one of the most utilised tools with UML support.

The application architecture (system architecture is understood as a global high-level structure whereby subsystems, the relationships among them, and the documentation thereof, are proposed [12]) is based on the Losavio's model for an object-oriented CASE environment [13]. This model coincides with other architectures for interactive applications, which promote independence between the semantics of the application and its interface. It proposes a three-tier architecture: *interface tier, semantic tier and integration tier.*

In the first version of ADAM CASE the two firsts tiers are clearly defined, although the integration tier is less outlined. Table 1 shows the functional elements of each tier, for the current ADAM CASE version.

Table 1. Tiers in ADAM CASE v1.0

Tier	Functionality
Interface	• Editing models
	• GUI
Semantics	• Analysis and design
	• Project management
Integration	• Export mechanisms for files

3.2 More outstanding functional characteristics

As stated before, the main goal of the present version of ADAM CASE is having a tool that allows the creation of DFDs and UML class diagrams, across the same GUI. Additionally, a set of facilities for analysis and design technical documentation is provided, emphasising the data dictionary management and the automatic report generation.

With the aim of presenting the most interesting parts of this tool, we are going to organise this subsection in two points: the DFD creation and the UML class diagram creation.

3.2.1 DFD creation

DFD is the most representative technique in structured methodologies for system functionality view modelling.

A DFD is created in ADAM CASE inside a structured project. The creation process is very intuitive and it is specially thought for levelled DFD, beginning in a context diagram and following a descending decomposition.

The creation of whatever diagram in a levelled DFD follows a communal steps, these are:

1. Add the components that form the diagram (processes, stores and external entities) and establish its properties. The components are chosen from the component tool bar (see Figure 2). Yourdon's notation has been selected for this version Yourdon [9].
2. Data flows are used to link the diagram components.
3. The data flows are associated with data dictionary entries.
4. If necessary, every process could be exploded to generate a new diagram.

The project monitoring and maintenance is easier with the use of the project navigator. It allows the direct access to the properties of each project element, both diagrams and diagram components.

One of the most interesting characteristics of this application is the data dictionary manager. This dictionary is constructed simultaneously with DFD development, making its modification and its integration in the final report easier.

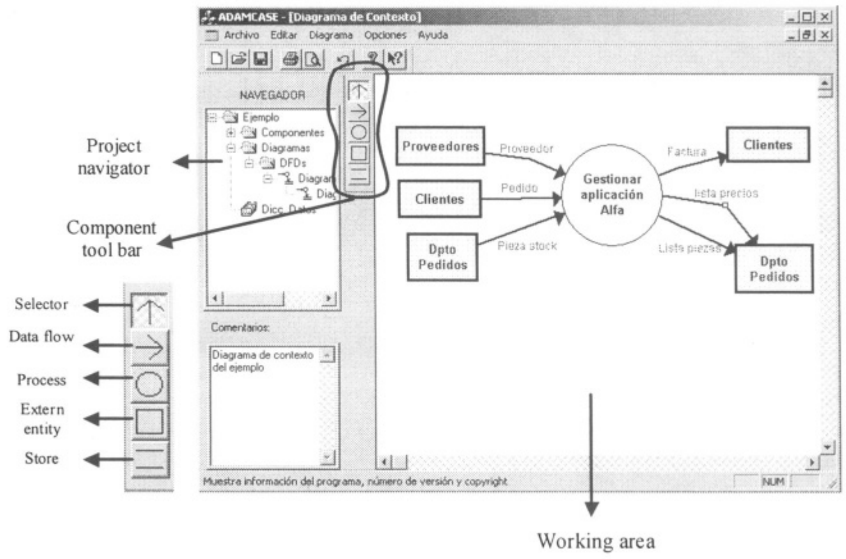
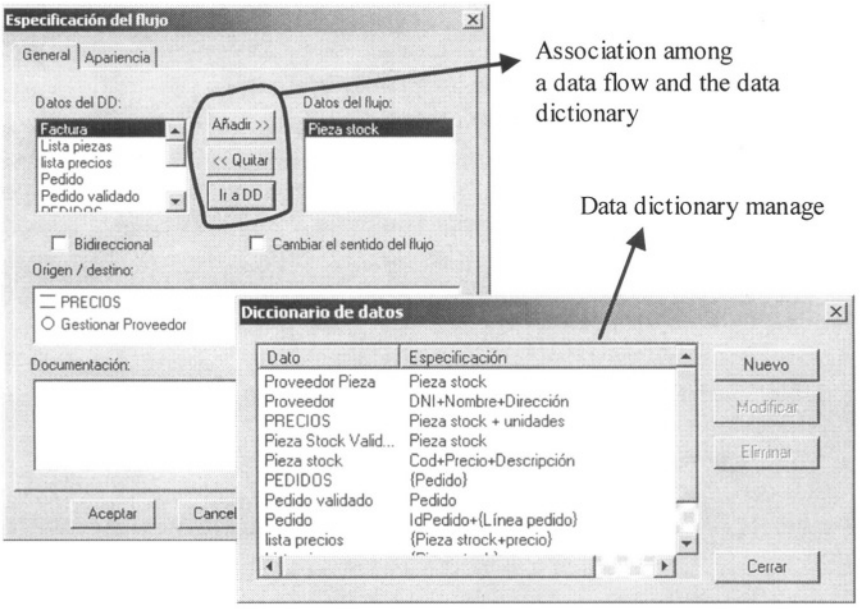


Figure 2. DFD creation

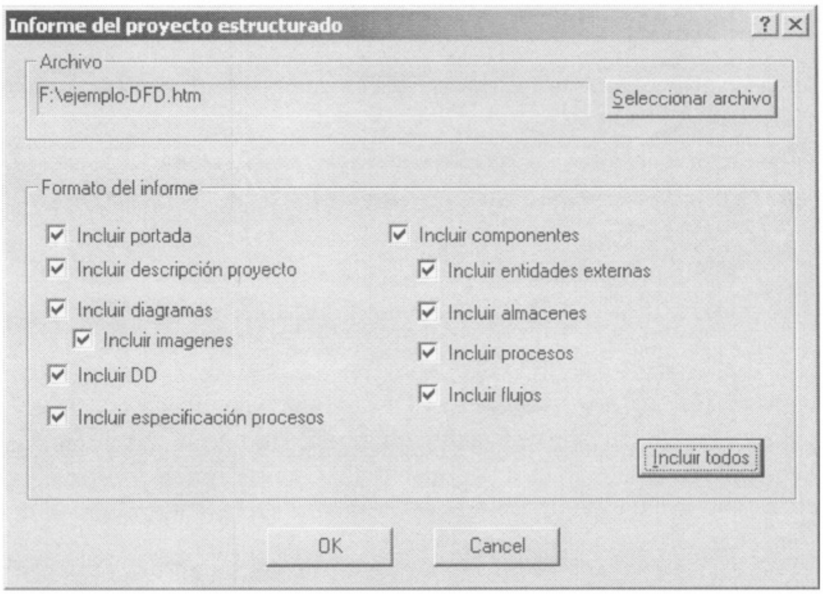
In Figure 3 the data flows specification dialogue box is presented. Here we can appreciate the tight relationship that exists among the data flow properties and a data dictionary entry. In this image the main dialogue box used for the data dictionary management is also included.



Dato	Especificación
Proveedor Pieza	Pieza stock
Proveedor	DNI+Nombre+Dirección
PRECIOS	Pieza stock + unidades
Pieza Stock Valid...	Pieza stock
Pieza stock	Cod+Precio+Descripción
PEDIDOS	{Pedido}
Pedido validado	Pedido
Pedido	IdPedido+{Línea pedido}
lista precios	{Pieza strock+precio}

Figure 3. Data dictionary management

When the DFD is finished, the tool has the option to generate the technical documentation with the system functional model in HTML format. In this process the user can select the elements he/she wants to introduce in the final report. The dialogue box to perform this option is shown in Figure 4.


Informe del proyecto estructurado

Archivo

F:\ejemplo-DFD.htm

Formato del informe

- [✓] Incluir portada
- [✓] Incluir descripción proyecto
- [✓] Incluir diagramas
  - [✓] Incluir imagenes
- [✓] Incluir DD
- [✓] Incluir especificación procesos
- [✓] Incluir componentes
  - [✓] Incluir entidades externas
  - [✓] Incluir almacenes
  - [✓] Incluir procesos
  - [✓] Incluir flujos

Figure 4. Functional model documentation generation based on DFDs

3.2.2 UML class diagrams creation

A class diagram mainly presents a set of classes and their relationships. This kind of diagrams is the most commonly found in object-oriented systems modelling. You use class diagrams to model the static view of a system [10].

A class diagram in ADAM CASE v1.0 is created inside a UML project.

There are not different levels in a class diagram, as we saw in the DFD, although package nesting is possible. There is not an explicit data dictionary, but in the generated documentation all the information related to the classes and their relationships is included.

The normal process to create a class diagram could be the next one:

1. Class creation, selecting its icon from the component tool bar.
2. Establishing its characteristics (attributes and methods).
3. Establishing the relationships among classes.
4. Refinement of the previous elements.

The user interface consists of the same components presented when DFDs were explained, i.e., there is a navigator, where everything about the project is collected; a component tool bar, with one icon for each main component in a UML class diagram; and a working area, where the diagrams are edited. These elements are shown in Figure 5.

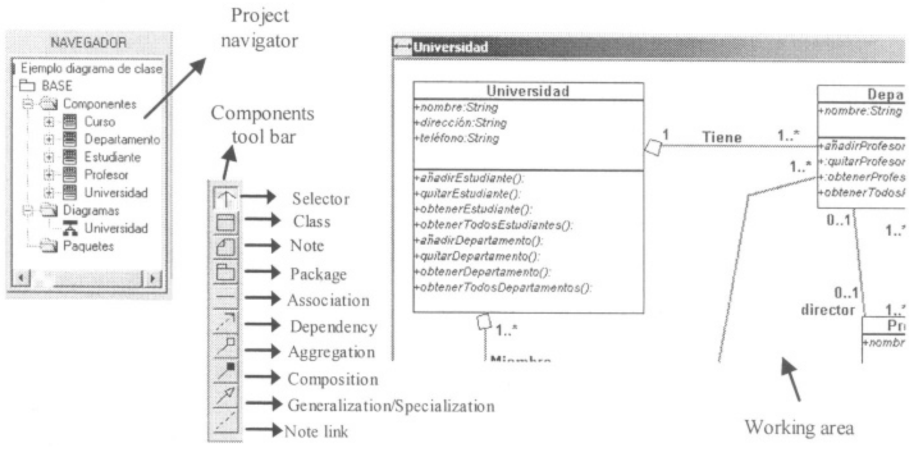


Figure 5. Main elements of the UML project graphical interface

Once the class diagram has been created, it is possible to automatically generate a technical report with the static view documentation in HTML format. As with the DFD, the user can select the elements that are going to be integrated in the final document.

4. RELATED WORKS

In the Tigris project [8] a UML diagrams edition tool is being developed, ArgoUML, under free software foundation licence.

ArgoUML is continuously under construction and it has support for more diagrams than ADAM CASE, but it is only centred on UML, regardless the structured methodologies and techniques.

5. CONCLUSIONS AND FUTURE WORK

In this paper ADAM CASE version 1.0 has been introduced. This tool has both DFD and UML class diagram creation capacity, all operating under the same user interface, thus making it easier to learn and manipulate.

This application is being used in Software Engineering and Object-Oriented Programming courses in the technical studies of Computer Science at the University of Salamanca in the academic year 2000-2001.

The most outstanding facilities in this first version, beyond offering both types of diagrams, are automatic data dictionary management and report generation in HTML format.

The future steps to be done are as follows:

- Increasing the number of supported diagrams. The most immediate ones will be entity-relationship diagrams and use cases.
- Porting the tool to another platforms, Linux in a first stage.
- Offering more possibilities in report generation, looking for integration with office automation applications.
- Using XML as support and interchange format of diagrams. This language could also be used to make a multilingual tool.

The current version of Adam CASE is available at http://tejo.usal.es/~fgarcia/docencia/isoftware/case/adamvl.htm

REFERENCES

1. Iivari J. Why are CASE tools not used? Communications of the ACM 1996; 39(10):94-103.
2. Sharma S., Rai A. CASE deployment in IS organizations. Communications of the ACM 2000; 43(1):80-88.
3. Granger M. J., Little J. C. Integrating CASE tools into the CS/CIS curriculum. Proceedings of the conference on integrating technology into computer science education, ITiCSE '96; 1996 June 2-6, Barcelona, Spain: ACM 1996: 130-132
4. Fuggetta A. A classification of CASE technology. IEEE Computer 1993; 26(12):25-38.
5. Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Ra-ma, 1996.
6. Pressman, R. S. Software Engineering: A Practitioner's Approach. 4th Edition. McGraw Hill, 1997.
7. Rational Rose Inc. Rational Rose Web Site. http://www.rational.com/products/rose/index.jtmpl. [Last time visited, July, 2000].
8. Tigris. ArgoUML Project main site. http://argouml.tigris.org/. [Last time visited, July, 2000].
9. Yourdon, E. Modern Structured Analysis. Prentice Hall, 1989.
10. Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. Object Technology Series. Addison-Wesley, 1999.
11. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. Pattern Oriented Software Architecture: A System of Patterns. John Wiley & Sons, 1996.
12. Sommerville, I. Software Engineering. 5th edition. Addison-Wesley, 1996.
13. Losavio F., Matteo A., Pérez M. An architecture for an object-oriented CASE environment. Journal of Object-Oriented Programming (JOOP) 1999; 12(6):49-54.

SICAS
Interactive system for algorithm development and simulation

Anabela Gomes* & António José Mendes**
**Instituto Superior de Engenharia de Coimbra and Centro de Informática e Sistemas da Universidade de Coimbra*
***Centro de Informática e Sistemas da Universidade de Coimbra*

Keywords: algorithm animation, software visualisation, educational technology, Programming teaching and learning

Abstract: The high levels of student failure in Programming classes suggest that traditional teaching strategies are not adequate for most students. In this work we present SICAS, a learning environment designed to help the student in basic Programming learning. With SICAS the student can design, simulate, test and compare algorithms for proposed problems. We also include a brief discussion about some of the features that we think are important for the success of an educational environment like SICAS.

1. INTRODUCTION

Programming learning and teaching is usually supported by traditional methods that predominantly use static materials. This does not seem appropriate, since programming essentially involves dynamic concepts. This is consistent with the high levels of failure common in Programming disciplines. Experience has shown that in an initial phase, many students have many difficulties in understanding and applying certain abstract programming concepts, such as control structures, in the creation of algorithms. We believe that the high abstraction level necessary in programming, the difficulties in problem solving techniques presented by many students, the need for a persistent practical work, so different of the type of study required by many other disciplines, constitute all important factors that contribute to this problem.

All the above difficulties originate some disorientation and lack of interest on many students, which demands a level of individualised support that teachers have difficulties to provide. Gomes and Mendes [1] present a summary of different systems proposed over the years to support Programming teaching and learning. Although the utilisation of some of those tools has produced positive results, the truth is that none of them is easily available or has a generalised use. So, we decided to design an educational environment, which we called SICAS (a Portuguese acronym to Interactive System for Algorithm Development and Simulation), with the aim to assist the teaching and learning of basic programming skills. SICAS does not include any expositive materials, but presents an environment that allows the students to develop their capabilities on the basis of the experimentation, helping them with the design, observation, analysis and visual simulation of algorithms, making it possible for them to detect eventual errors, correct them and learn from those activities. We believe that such activities will improve the student's problem solving capabilities.

2. SICAS CHARACTERISTICS

The success of a learning environment depends not only on its functionalities and potentialities, but also on the way it interacts and communicates with the student. During SICAS development, there was a constant concern with interaction aspects. The idea was to avoid distracting features, so that the students can focus on the main objective of the program. We tried to design SICAS with the following features in mind:

– Be visually organised, in order to motivate the students' habits of organisation and systematisation of the tasks, avoiding dispersion and distraction. "...a clear visual organisation is essential for an effective communication..." [2].
– Be simple, obvious and intuitive, requiring little time to master. The interface should be organised in order to be easy to browse [3].
– Contain only essential information, so that the students can concentrate on fundamental tasks and not to get distracted by superfluous details or redundant information. Smiths and Mosier [4] also consider that all the information that the user needs, and only that information, should be available.
– Be adaptable, in order to accommodate user preferences, as well as the addition, removal and edition of examples, proposed exercises and their solutions. We believe that it is desirable to give some degree of freedom to the user, but we should consider that sometimes the existence of too

many alternatives can devaluate the importance of the act of choosing and cause frustration instead of satisfaction [5].

– Be powerful, giving the sensation of control to the student and consequently provide a positive and motivating effect. However, we think that the control should not be absolute, so that the environment can support learning more effectively [6].
– Have a set of alternative representations for the same problem, providing the students with different perspectives and styles of reasoning in algorithm development and analysis.
– Enable the visualisation, animation and simulation of student designed algorithms, in order to facilitate the comprehension of the underlying concepts, favouring the understanding of this type of matters [7].
– Be retroactive and interactive, generating appropriate feedback to the students, supplying information and suggestions that help them to discover, understand and correct errors that may exist in their solutions [8]·

3. SICAS DESCRIPTION

SICAS includes support to two types of activities: design/edition of solutions (algorithms) to the problems proposed and execution/simulation of those solutions. In the first case, the user can construct algorithms using visual representations, flowcharts. They can use graphical symbols that represent the structures necessary to design the algorithm. In the second case, the student can simulate and animate the execution of the algorithms she/he designed, analysing in detail and at the desired pace the various phases and entities involved.

SICAS includes a teacher mode that allows the creation and edition of a library of suggested problems. A problem is composed by a statement that describes it, one or more algorithms that solve it and, eventually, a solution (defined by the teacher) with some data test sets. The idea of the existence of several algorithms for the same problem is to provide different types of reasoning and to support diverse forms of understanding, allowing the student to compare them and to choose which resolution strategy presented is more adequate for the problem. The data test sets allow the student to easily verify if his/her algorithm really solves the problem.

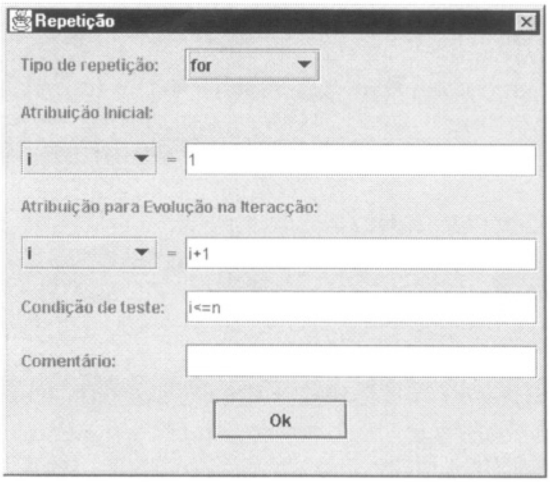
In any of the working modes (student or teacher) the user can consult or edit one of the algorithms. If the user is in the teacher mode she/he can still open an algorithm built by a particular student. If the user is in the student mode he/she can select one of their algorithms or one of the algorithms that the teacher may have created for the same problem.

The algorithm design is supported by an iconic environment where the user can build a flowchart that represents her/his ideas. This representation mode was chosen after a deep reflection. We decided to privilege the use of graphical representations (flowcharts), instead of verbal specifications (pseudo-code), not only because the results of several studies we have analysed, but also because we believe that this form of representation is more appealing (capturing the attention of students more), takes advantage of the potential of the human visual system more to facilitate understanding, and is simpler and probably less prone to errors. There are many studies that evidence the valuable contribution of structured flowcharts relative to pseudo-code for algorithms understanding. For example, Scanlan [9] carried through an experience where he concluded that flowcharts:

– Take less time to understand;
– Produce less understanding errors;
– Give more confidence to the students on their understanding of the algorithm;
– Reduce the time necessary to answer questions related to the algorithm;
– Reduce the number of times that the students need to analyse the algorithm.

The structures the students can use when creating flowcharts in SICAS are:

– Attribution - It sets the value of a variable with the result of one expression.
– Input/Output - It makes it possible to read values from the user and to write in the output window.
– Repetition - It allows the repetitive execution of some action. Figure 1 shows a dialogue box where the student can specify the repetition characteristics.
– Selection - It allows choosing between two sets of actions that may be executed.



Tipo de repetição: for

Atribuição Inicial:

i = 1

Atribuição para Evolução na Iteracção:

i = i+1

Condição de teste: i<=n

Comentário:

Figure 1. Repetition specification

The dialogue boxes used were designed to include only the minimum set of information necessary to execute the corresponding programming structure. This avoids syntax errors common in novice programmers and makes them think more carefully about the different components of each structure.

At any moment, the student can easily delete, modify or copy any component. This can be done through menus, the corresponding icons in the toolbar or using activating context menus using the right mouse button. In order to guide the student as much as possible, the colour of the selected component is changed to point out this fact.

The lines connecting the components are automatically inserted by the system, avoiding inconsistencies that students might introduce in the flowchart.

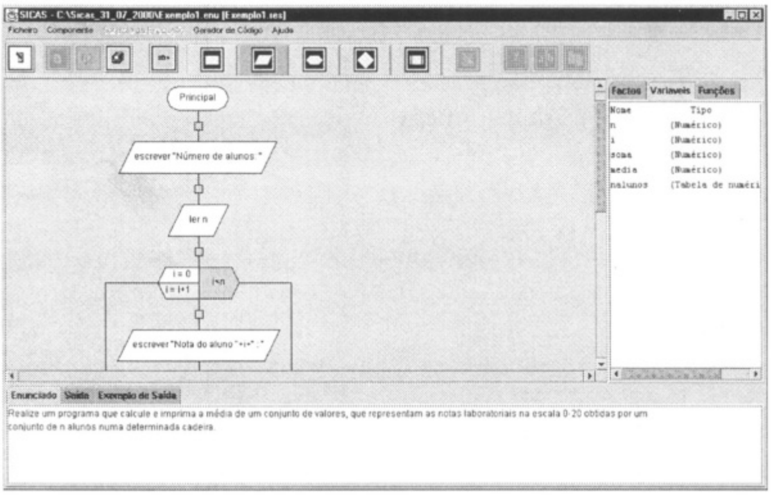
SICAS only supports the use of four data types: “Numbers”, “Strings”, “Arrays of strings” and “Arrays of numbers”. As SICAS main objective is to support basic Programming learning these data types seem enough. We consider that the diversity of data types, although an important aspect of programming languages, usually constitutes a source of increased difficulty distracting novice Programming students from algorithm design.

The construction of expressions in SICAS uses a syntax similar to C and JAVA syntaxes. This decision took into consideration that the environment potential user will program, in a later stage, in one of those languages. However, we tried to minimise syntactic details, so that the student can concentrate completely in the main task, the creation of an algorithm to solve a problem.

Another important feature is the support for student defined functions. This tries to introduce the students to the concept of modularisation from the beginning of Programming learning. Modularisation has several advantages, such as:

- Making the algorithm more legible and easy to understand.
- Helping to manage and solve more complex and/or bigger problems.
- Allowing the alteration of a module or function without changing the rest of the algorithm.
- Avoiding redundancy, making the reduction of the number of components possible.

The construction of functions is done using the above mentioned structures and according to the rules and options previously described for any resolution. SICAS also have a set of pre-defined functions that can be used in the construction of expressions without the need of its declaration. There are functions for the manipulation of numbers (example: min, sqrt, os...) and strings (example: ascii, length, toLowerCase...). Figure 2 shows some of the structures mentioned above used in the design of an example algorithm.



Nome	Tipo
n	(Numérico)
i	(Numérico)
soma	(Numérico)
media	(Numérico)
nalunos	(Tabela de numéri

Realize um programa que calcule e imprima a média de um conjunto de valores, que representam as notas laboratoriais na escala 0-20 obtidas por um conjunto de n alunos numa determinada cadeira.

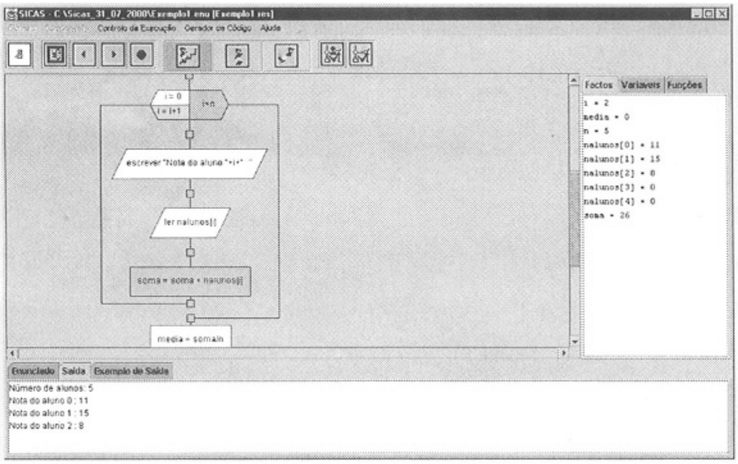
Figure 2 – Example of algorithm design

Any algorithm created with SICAS can be automatically translated to pseudocode, C or JAVA. These various alternatives are not related with their importance, but are a way to show the students that a well designed algorithm can be easily translated to several languages. So, the most important thing is algorithm design and not in which programming language it will be implemented and the corresponding syntax details.

After building an algorithm to solve a problem, the student can see its animated simulation. The student can have some control over the simulation, namely:

– Control over the rhythm at which the simulation progresses, going step-by-step between instructions, with a continuous but slow execution or with a faster uninterrupted execution.
– Pausing the simulation, allowing a deeper analysis of available data (e.g. the value of some variable) and/or a discussion with the teacher or other learners.
– The possibility to go back in the execution, so that the student can repeat some less understood part of the execution.

During the simulation the component in execution is highlighted with a different colour, so that the student can easily locate what is happening. Figure 3 shows an aspect of SICAS during an algorithm simulation.



```
i = 2
media = 0
n = 5
nalunos[0] = 11
nalunos[1] = 15
nalunos[2] = 8
nalunos[3] = 0
nalunos[4] = 0
soma = 26
```

```
Número de alunos: 5
Nota do aluno 0 : 11
Nota do aluno 1 : 15
Nota do aluno 2 : 8
```

Figure 3 – Algorithm simulation

If the teacher has given a set of test data (inputs and corresponding expected outputs of the algorithm) for the problem being solved, the student can also test her/his own algorithm with the same data, hence validating (or not validating) it. This is particularly useful if the teacher specifies special cases in the input data (e.g. invalid data or data that will likely produce error situations), since many students tend to get satisfied if their solutions work for an average case in the possible input data, without caring to see if it still works in other cases.

4. CONCLUSION

SICAS is an educational environment that intends to help students to understand and master the utilisation of basic programming structures. The student can design algorithms that solve programming problems possibly proposed by a teacher. The system uses an interactive graphical interface, where the basic programming structures are represented through standard symbols.

This environment can be used by students individually or in group, in classes or in self study. This can allow weaker students, often with difficulties to cope with the class pace, to improve their programming knowledge and experience.

The first prototype version of SICAS is currently being evaluated by a group of experienced Programming teachers and also with small groups of students. We hope that a good number of improvement suggestions will result from this work. A revised version will be produced and a larger scale evaluation will be realised in the near future.

REFERENCES

1. Gomes, A. e Mendes, A. J. (1999) *A animação na aprendizagem de conceitos básicos de programação.* Revista de Enseñanza y Tecnología, N° 13, pp. 22-32.
2. Mullet, K. e Sano, D. (1995). *Designing Visual Interfaces: Communication Oriented Techniques.* Englewood Cliffs: SunSoft Press & Prentice Hall.
3. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. e Carey, T. (1994). *Human-Computer Interaction.*.Wokingham: Addison-Wesley Publishing Company.
4. Smith, S. e Mosier, J. (1986). *Guidelines for Designing User Interface Software.* Report ESD-TR-86-278. Bedford. MA: MITRE Corporation.
5. Miller, G. (1956). *The magical number seven, plus or minus two: some limits on our capacity for processing information.* Psychological Review, N° 63.
6. Gentner, D. (1992). *Interfaces for Learning: motivation and the locus of control.* In F. Engel, D. Bouwhuis, T. Bösser, G. d'Ydewalle (Eds.), "Cognitive Modelling and Environments in Language Learning", Spring-Verlag.
7. Lawrence, A., Badre, A., e Stasko, J. (1994). *Empirically Evaluating the Use of Animations to Teach Algorithms.* In Proceedings of the 1994 IEEE Symposium on Visual Languages. St. Louis, pp. 48-54.
8. Bangert-Drowns, R., Kulik, C., Kulik, J. e Morgan, M. (1991). *The Instructional Effect of Feedback in Test-Like Events.* Review of Educational Research, Vol. 61, N° 2, pp. 213-238.
9. Scanlan, D. (1989). *Structured Flowcharts Outperform Pseudocode: An Experimental Comparison.* IEEE Software, Vol. 6, N° *5,* pp. 28-36.

Simurob and JRF

Teaching tools for robot simulation and programming

J.L. Gómez, I. Álvarez, F.J. Blanco, F.J. García, B. Curto
Dpto. Informática y Automática, Universidad de Salamanca.
(inavia, fgarcia@gugu.usal.es, jblanco, control@abedul.usal.es)

Key words: Robotics, Simulation, 3D Graphics, Java

Abstract: This is a set of tools to be used with teaching purposes in a robotic environment. A first tool is a graphic simulator of an educational robot, the MA 2000 by Tecquipment, which allows the display of both the robotic arm and the obstacles present in its environment. These can be built from within the simulator in a simple way, by means of a series of primitives that can be used to produce a realistic-looking work environment. It is also possible to program sequences of arm movements by means of a keypad, which is analogous to the one used in the actual robot. The sequences thus produced may be executed directly by the robot.
A second tool has been developed in order to offer a better flexibility; it was our purpose to make it available in just about any computing environment. It allows the student to build and simulate the behaviour of complex robots within a fully configurable working environment. Robots are built simply; there is full freedom concerning the number and type of articulations, and also about the shape of the elements of which they consist of. Furthermore, we have included the possibility of generating sequences of movements. Both the pieces of the robots and the obstacles in their environment are built by means of VRML files, which makes it possible to produce very realistic-looking robots and environments.

1. INTRODUCTION

The advantages of computer simulation are well known; this is especially true when using costly equipment, which is complex to use and possibly dangerous in execution. Low-cost training and operational safety are probably the most important benefits of computer simulation. As far as the teaching of robotics is concerned, these techniques are especially interesting since they offer the student a risk-free training [1, 2, 3, 4].

This paper describes two simulation tools for robotic environments, which we use within a postgraduate course, “Sistemas avanzados de la producción”, which belongs to the Doctoral Program of the Departamento de Informática y Automática at the University of Salamanca. This course introduces the student to, among other subjects, two of the main aspects of robotic manipulation: the movement and programming of a robot.

In order to get a clear vision of the practical aspects involved, it is essential for the student to be able to work directly with real robots. However, the high costs of acquisition and maintenance of robots preclude this approach. In most cases, industrial robots are abandoned in favour of teaching robots: such is the case of our department, since teaching robots are cheaper to acquire and to maintain. This, of course, means a reduction in their functional and programming characteristics. Unfortunately, this does not mean one can leave this kind of robots in the hands of inexperienced students, since it would give rise to many problems and risk situations (both personal and material) of the kinds one can find in a robotics lab.

A clear alternative, as mentioned before, is to have the student trained previously by means of software simulation, in such a way that he or she can become used to the robot, its behaviour, its possible movements and its programming. A graphic simulator should be as accurate as possible in its movements and programming, since this will let the student practice with sequences of movements before trying them out on the real robot. Thus, we can be sure that no dangerous situations can arise, such as collisions of the robot with its environment. Hence, the first tool we introduce, Simurob: it allows us to simulate the movements and programming of an educational robot of the PUMA type, the MA 2000 by Tecquipment [5]. Our simulated environment accurately reproduces both the robot and the way it is programmed.

However, in an advanced course (such as ours) it is very important to let our students practise with different kinds of robots: this will give them a better knowledge of the various and complex robotic structures they will find in industrial environments [6]. There is yet on more aspect we must take into account: the possibility of both studying and designing specific robots for various tasks in complex environments.

Since we cannot possibly have multiple robotic structures (with various degrees of freedom, different kinds of articulations, etc.) and we do not own a sufficient number of pieces of tools to build our own robots, the only way we can deal with this sort of student practice is using simulation. Thus, we introduce a second tool, JRF (Java Robot Factory), which allows us to create just about any kind of robotic structure, as well as to execute movements and programme this structure. This is done by means of a fully generic interface, without discarding the possible use of configurable environments.

This paper consists of two parts, which describe these tools. First of all, we shall describe Simurob, describing the most relevant aspects of its functionality. Afterwards, we shall describe JRF and we will finally offer our conclusions.

2. SIMUROB

The main purpose of this simulator is to accurately display the behaviour of the real robot (MA 2000) both in its movements and in aspects related to programming its sequences. The sequences we build by means of the simulator can be directly loaded and executed by the robot's software. This lets the student become acquainted with the robot before actually working with it.

The functionality offered by the simulator is as follows:

– It displays a graphic simulation of the robot within its working environment.
– It creates sequences of movements that can be saved as files and then loaded and executed by the real robot.
– It creates working spaces that may be saved for later simulations.
– It shows various numeric values, which can be of use both for the simulation itself and for the user to better understand what he or she is simulating.

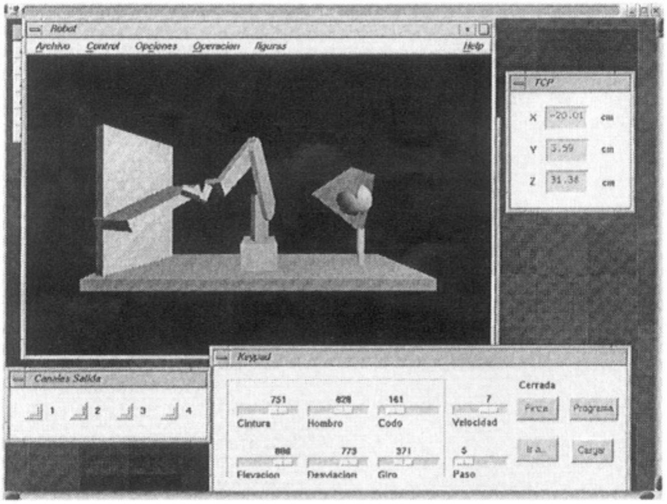
The simulator was built for the SGI Indy platform, leveraging the graphic potential offered by these machines. To be precise, we have used the OpenGL library [7] for scene creation, and the IrisViewkit class library for the user interface, in order to make it as intuitive as possible. Of course, these tools reduce portability to other platforms, but the performance obtained compensates for this inconvenience. Our students boot the simulator on an SGI Origin server and the display is done on any MS Windows platform by means of an X server which supports OpenGL, such as Exceed 3D.

2.1 General aspects

Figure 1 shows the application while in execution. The following main elements can be distinguished:

– A main window, where scenes are displayed. It shows a menu that lets us access the full functionality of the simulator.

- The Keypad, a dialogue box that allows us to control the movements of the articulations as well as programming sequences.
- The TCP (Tool Central Point), which shows the position of the centre of the wrist in XYZ co-ordinates. These are relative to the size of the real robot, which lets the student have a better idea of the wrist's real position.
- Output Channels, which shows the state of the robot's four output channels. These are used for synchronisation purposes with other elements of the working environment.



X -20.01 cm
Y 3.59 cm
Z 31.34 cm
751 Cintura
828 Hombro
161 Codo
7 Velocidad
886 Flevacion
773 Desviacion
371 Giro
5 Paso

Figure 1 Simurob main window.

The main window allows us to access the full functionality of the simulator through menu options; it is also in charge of showing the scenes and the simulation of movements of a robot that is thoroughly similar to the MA 2000.

2.2 Sequence programming and execution

The File *("Archivo")* menu allows us to access all of the options related with the creation of sequences; it also allows us to create sequences, to load an existing sequence and to save the active sequence to a file.

Sequence structure is fully analogous with the one used by MA2000's control software; however, this is hidden from the user, leaving him or her free to indicate the movements that must be executed by the robot. When a new sequence is created, the Keypad appears automatically, thus allowing us to control the robot's movements. From the Keypad, one can configure each step, by specifying different values related to the various articulations, movement speeds and, in the case of a control step, it is possible to configure the parameters by means of a button-controlled dialogue box.

Once we have the desired sequence, we can launch the simulation of the associated movements. Hence the user can check the robot's movements in order to see whether they are executed as expected, with no collisions. Execution may be stopped at any time. During execution, the Keypad will show the articulation values and the status of the actuator for the step being executed.

2.3 Creation of work spaces

About the work space, the user can create, save and open different environments, which will consist of simple geometric figures such as quadrangular prisms, spheres, cones, etc.

When creating a environment, it is possible to generate realistic-looking scenes by putting in place several of these geometric figures. Once the figure to be placed has been selected, one can change its size about the three possible axes. There are three further scrolling bars that allow us to place the object in its proper position within the environment.

3. JRF (JAVA ROBOT FACTORY)

As stated above, this simulator makes it possible both to create robotic structures and to program sequences of movements for them. This allows the student to work (albeit in simulation mode) with robots as complex as the ones that he or she will find in industrial environments. In this way, the student may perform the cinematic study of various robotic structures. In the same way, it is possible to find the most advantageous robot configurations, depending on the working environment and on the task to be carried out.

The functionality offered by the simulator may be summarised as follows:

- It offers a tool for the design and generation of robots.
- It offers a graphic simulation of the robots and of the working environment within which they will move.
- It creates and executes sequences of movements that may be saved on file for further reference.

The simulator has been developed on a PC-compatible machine, and runs both on Windows and on Linux. Java has been used as the programming language, given its ability to run on multiple platforms. We have used the API Java3D API [8] for the 3D graphics that comprise the robot and its working environment. Students can execute the simulator on a PC-compatible machine running Windows with OpenGL support.

3.1 General aspects

This application can be executed in two well-differentiated modes; it is possible to change modes when it is running. The execution modes are as follows:

- Robot creation: The main window offers the tools needed for the design of various structures.
- Display of the robot and its motion: The main window runs in simulation mode, thus allowing both a direct interaction with the robot and the generation and execution of sequences.

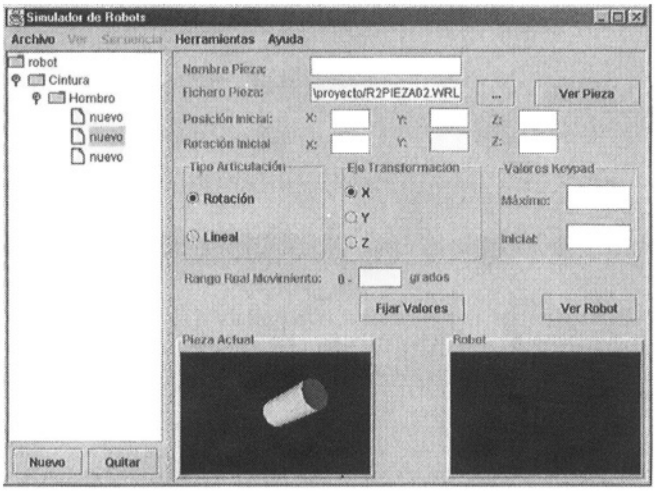
There are several additional elements needed for the various tasks:

- A movement control panel, which allows us to produce the movements of the articulations.
- The programming environment, a dialogue box that makes it possible to generate complex movement sequences by using a set of simple steps.
- Input/Output channels, which show the status of the robot's interface. They are used for synchronisation purposes with other elements within the working environment.

3.2 Robot creation mode

One to this application's most important features, and one that offers great versatility, is its Design Environment. This environment is the one used to create a customised robotic structure.

As shown in Figure 2, the window consists of three basic areas. The one on the left contains a directory tree for the robot structure; the upper right zone is used to configure part parameters and on the lower right, we find two windows for displaying 3D objects.



Fichero Pieza: \proyecto/R2PIEZA02.WRL

Tipo Articulación: [✓ Rotación] Lineal

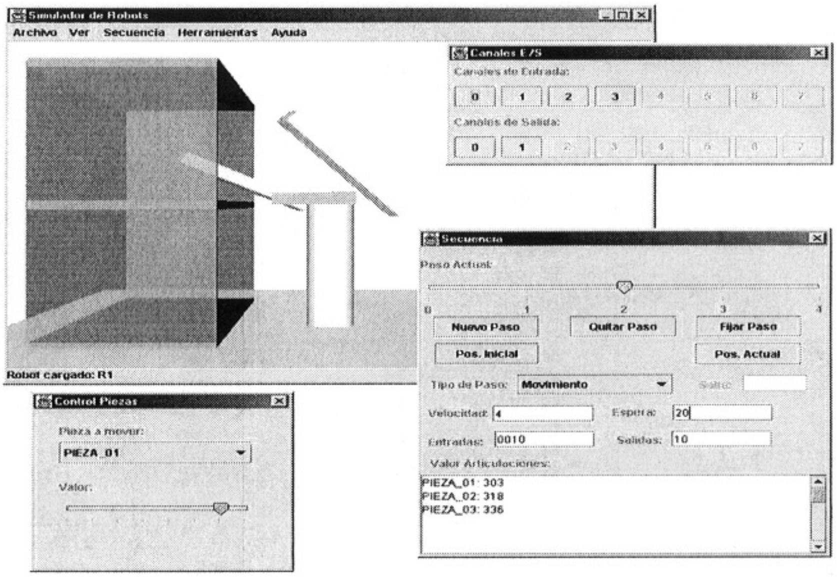
Eje Transformacion: [✓ X] Y Z

Figure 2. JRF Design Environment

Hence, in order to create a robot it suffices to create a tree structure that describes the placement of the various elements it consists of. Afterwards, each part must be assigned its corresponding configuration values: these will include the name to be used when referring to the part, its position relative to the previous-level part, the type of articulation, etc. One of the more interesting parameters is the one that specifies a "Part file", which is a VRML [9] that describes the geometric shape of the part and its texture.

In this way, by having access to a wide library of geometric shapes, one can build almost any type of articulated robotic structure, no matter how complex. It must be taken into account that VRML is a standard for 3D presentations, which makes it possible to easily create any library that one may need.

Once we have designed the robot we shall work with, it can be saved for later use.



Robot cargado: R1

Pieza a mover: PIEZA_01

Tipo de Paso: Movimiento

Velocidad: 4 Espera: 20

Entradas: 0010 Salidas: 10

Valor Articulaciones:

```
PIEZA_01: 303
PIEZA_02: 318
PIEZA_03: 336
```

Figure 3: JRF Simulation Environment.

3.3 Simulation Mode

When in this mode, the main window looks like Figure 3. The options in File menu (*"Archivo")* make it possible to open both the previously-generated robots and the working environment described in a VRML file.

Interaction with the robot is also possible in this mode, both by moving any of its parts through the Keypad and by executing a sequence of movements.

The sequence editor further allows us to generate sequences as a set of steps, in a way that is fully analogous to Simurob's. Steps can be of two types:

– Movement: for configuring the target position, speed, waiting time and continuation condition.
– Conditional branch: Depending on input values, the next step will be taken or the robot will branch to another step within the sequence.

4. CONCLUSIONS

This paper introduces two software tools that may be used for robotics teaching. They make use of advanced graphics environments for the simulation both of robots and of their working environments. The student can achieve a reasonable degree of understanding in different aspects of robotics: robot programming, cinematic calculations, obstacle avoidance, etc.

The first tool, Simurob, lets the student work with a very accurate simulation of an educational robot, the MA2000. The tool allows reproduction of working environments.

A second tool, JRF, lets the student experiment with a wide range of robotic structures; it is possible to build fully customised robots. The tools offer the possibility of navigating within the working space in order to achieve any point of view.

In conclusion, this set of tools are of special interest since they complement each other and may provide a safe and intuitive platform for the teaching of different aspects of robotics.

REFERENCES

1. Ivan A. Navia, A.F. Zazo-Rodríguez, M. Zorita-López, "SIMUROB: A Graphic Simulator for Robotic Sequence Programming", Proceedings of the IASTED International Conference, 1994, pp. 36- 39
2. Tzivi Raz, "Graphics Robot Simulator for Teaching Introductory Robotics", IEEE Transactions on Education, Vol.32, No.2, May 1989, pp. 153-159
3. Alan Watt, "3D Computer Graphics", Addison-Wesley, 2^a e4d., 1993
4. Robert B. White, R.K. Read, M.W. Koch, R.J. Schilling, "A Graphics Simulator for a Robotic Arm", IEEE Transactions on Education, Vol. 32, No. 4, November 1989, pp. 417-429.
5. TecQuiment, "MA 2000 Robot; Operator´s Manual", TQ.
6. Mark W. Spong y M. Vidyasagar, "Robot Dynamics and Control", John Wiley & Sons, 1989.
7. Mark Segal, Kurt Akeley, "The Open GL® Graphics System: A Specification (Version 1.2)" Silicon Graphics, Inc., 1998
8. Sun Microsystems, "The Java 3D™ API Specification, Version 1.2", April 2000
9. Rick Carey and Gavin Bell, "The Annotated VRML 97 Reference", Addison-Wesley, 1997-1999

Collares Ortofónicos

System for the training of the suprasegmental parameters of intensity and rhythm of the articulated sound

F. J. González Cañete, S. Torres Monreal, R. Conejo Muñoz
*equinoxe@dte.uma.es *, monreal@uma.es **, conejo@lcc.uma.es **
**Departamento de Lenguajes y Ciencias de la Computación*
***Departamento de Psicología Básica*
Universidad de Málaga, E.T.S.I. Informática;
Apartado de correos 4114, Málaga. Tel. 952 132844.

Key words: Interactive learning environment. Speech suprasegmental parameters. Cognitive-linguistic processes. Speech and language intervention.

Abstract: This article describes the structure and working of an interactive multimedia environment for the training on two of the speech suprasegmental parameters (loudness and rhythm) for its application in many types of speech disorders and cognitive-linguistic pathologies. The environment allows the tutor to configure and personalise the exercises to treat specific cases or to let the proper environment propose the exercises following various configured parameters and, after that, analyse the results obtained by the clients.

1. INTRODUCTION

Collares Ortofónicos is a computer application aimed at the logopedic treatment of children with perception troubles of rhythm, loudness or both parameters in the processing of the acoustic signal related with speech [1], that is, the bases or the articulated sound. Among its possible users, we could mention those people affected with profound deafness and other people with a deficit in the space-time processing of the articulated sounds of speech, rhythm, sequence of figures and sounds, simultaneous versus successive processing [2,3], attention deficits, among other impairments [4].

The clinicians and language therapists, in their task with hearing impaired children, must get these children to distinguish the speech suprasegmental parameters, that is, sound, loudness, voicing onset, duration, prosody, rhythm and pause. These parameters have been usually learned by hearing impaired children with imitation and lip-reading imitation, or with discrimination strategies based on selection techniques of frequency bands and hearing stimulation [5]. "Collares Ortofónicos" pretends that children could have an entertaining autocorrection, mainly when the computer output is connected with the SUVAG amplifier (Sistema Universal Verbal Auditivo Guberina) and the deaf people can perceive their emissions both by lip-reading and by vibrotactic perception, with the vivag-suvag vibrators, and the headphones adapted for the stimulation of their hearing rests.

The idea of associating a colour to each sound and the characteristics of various geometric figures to various articulated sound properties, that is: the size to indicate intensity and the shape to indicate the duration, have allowed us to work, with success, with babies and children younger than three years, some basic linguistic aspects. It is well known that babies and children have an exceptional predisposition to acquire linguistic aspects of great complexity [6, 7, 8, 9]. This process used to be done with a box or a bag with various shapes, sizes and colour figures. The methodology is to prepare a physical element (sphere or cylinder), to articulate the corresponding sound and to present it in a fast way during 1 second more or less, so the child could associate it with the hearing stimulus. In this way, figures are shown in contrast position or phonological opposition so that the child can extract, unconsciously, the phonetic characteristics of the different speech sounds. After the presentation phase, that is, to attract the child's attention, we have the imitation phase and finally the minimum pair discrimination phase.

"Collares Ortofónicos" follows, essentially, the design on which we have been working with manageable materials. They are sequences of figures associated with phonetic characteristics, based on phonological oppositions, joined with a little imaginary rope to impose an established order in the execution of the exercise. The sequence will follow the left-right and up-down order to create an automatism that will facilitate the learning of the reading process. The result of each screen is a ring, and each figure is a piece of the ring, that represents two or three parameters of speech sounds (MOC [10], for material, task and procedure revision).

"Collares ortofónicos" is going to join the list of commercial applications that have been developed to reinforce, rehabilitate, and study the speaking perception and production as psycholinguistic processes. Among these applications we can emphasise its use among the special education teachers and language therapists: (SpeechViewer-III of IBM [11] and the VISHA system [12].

2. SYSTEM DESCRIPTION

The application, from an organisational point of view, has two modules or scenarios, one of them is oriented towards the teacher and the other towards the pupil, both integrated in the same program. The teacher module function is to facilitate the creation and checking of the exercises that are going to be proposed to the student. On the other hand, the pupil's module is oriented towards the making of exercises and towards the calculation of success and error statistics.

The application, using only vowel sounds, follows a preestablished sequence for the phonological training: each pupil must do 27 predetermined exercises, where the five Spanish phonologic vowel oppositions are treated: 1) open-close, 2) loud-soft, 3) long-short, 4) front-back and 5) round-enlarged. These exercises are proposed automatically and progressively by the system. When the pupil has finished these exercises, he/she could do other exercises developed by the teacher for specific purposes. The system can generate exercises automatically following various preconfigured parameters selected by the teacher. In this way, the program can adapt itself to the concrete needs of each pupil. To make the decision of what generation parameters are going to be selected, the teacher will base himself/herself on the progress statistics and reports generated by the system.

3. TYPES OF EXERCISES

3.1 Understanding exercises

They are composed of a sequence of figures with a phonetic meaning, that is, they represent phonetic characteristics of sounds. The pupil must read or articulate the figures to do the exercise. These exercises are named this way because the pupil has to *understand* and learn the phonetic meaning of the figure accurately.

– Colour	– Vocalic phonemes (each vowel is associated to a colour)
– Blue	– /a/
– Green	– /e/
– Yellow	– /i/
– Red	– /o/
– Black	– /u/
– Shape	– Sound duration
– Sphere	– Short

- Size – Sound intensity
  - Big – Strong
  - Small – Weak

A sequence of figures is presented on the screen to the pupil, where the characteristics and meanings are shown beforehand, and he/she has to pronounce the phoneme that corresponds to each of the figures, into a microphone.

The pupil will have to pronounce the sound corresponding to the current figure, which is the one that has an arrow inside (figure 1).

While the exercise is being done, there will be two indicators, one of them on the left and the other at the bottom of the screen. They are the intensity and duration indicators, respectively, of the sound pronounced by the pupil.

If the sound of the figure is pronounced correctly, a little smiley face will appear in the position of the arrow, a success sound will be heard (positive reinforcement), and the exercise will continue to use the next figure. If the pronounced sound is not the correct one, a little sad face and a failure sound will be heard (negative reinforcement), so the pupil will have to repeat the pronunciation of the figure.

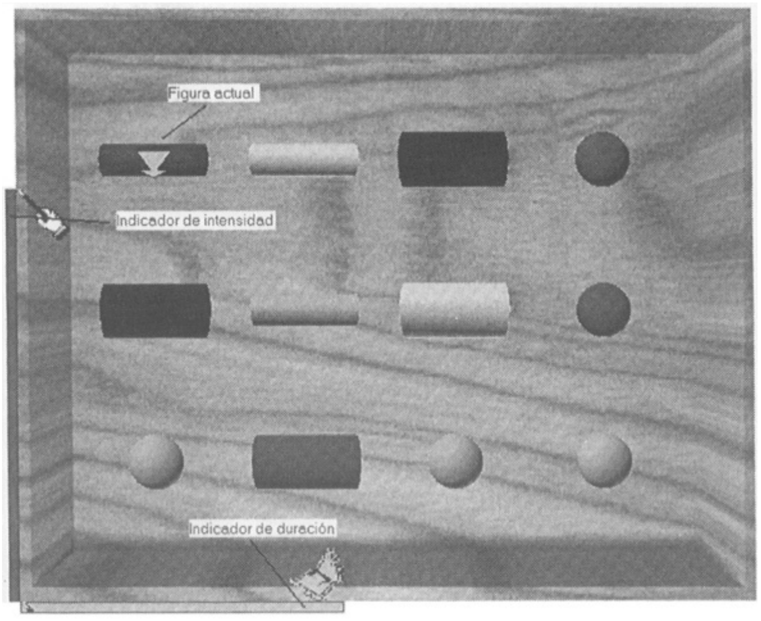


Figure 1 Understanding exercise

When the last figure of the exercise is correctly pronounced, a congratulation sound will be heard and a dancing cookie will appear as a positive reinforcement.

3.2 Composition exercises

They are a series of letters (vowels) followed by the corresponding sound, and the pupil must choose the corresponding figure (as in the understanding exercises).

– The letters are the five Spanish vowels (a, e, i, o, u).

– Size of the letters	– Intensity
– Capital letter	– Strong.
– Small letter	– Weak.
– Shadow	– Duration
– Vowel with shadow	– Long.
– Vowel without shadow	– Short.

The pupil will have to propose the corresponding figure to the letter to answer. Before that, the corresponding sound of the current letter will be heard.

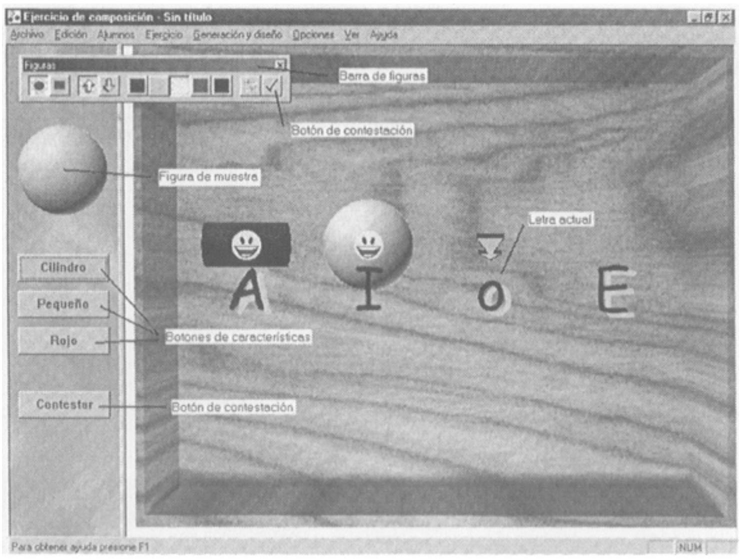

Ejercicio de composición - Sin título
Archivo Edición Alumnos Ejercicio Generación y diseño Opciones Ver Ayuda
Figuras
Barra de figuras
Botón de contestación
Figura de muestra
Letra actual
A
I
O
E
Cilindro
Pequeño
Rojo
Botones de características
Contestar
Botón de contestación
Para obtener ayuda presione F1
NUM

Figure 2. Composition exercises

This kind of exercise tries to help the hearing impaired child *hear* the sound produced by the computer connected to the S.U.V.A.G. The suvag is a system for the amplification and modification of the speaking signals. It has a vibrotactil and auditive output that allows the hearing impaired person to perceive the stimulations, whatever their age and evolutionary or linguistic development.

While the exercise is being done (figure 2), the figure to answer will be indicated by an arrow that will be on top of the figure. The size, shape and colour parameters must be changed to adjust them to the characteristics of the figure that the pupil wants to answer.

When the figure has the desired characteristics, the pupil can answer by clicking the answering button.

3.3 Onomatopoeia

They are pre-programmed comprehension exercises that need the articulation of the onomatopoeia of an animal, object or action sound. The onomatopoeias are short speaking segments, normally with an accepted phonological syllable. These phonologic segments have a meaning that the child extracts by associating the images with the corresponding onomatopoeias. The onomatopoeias are the second phase in the logopedic rehabilitation.

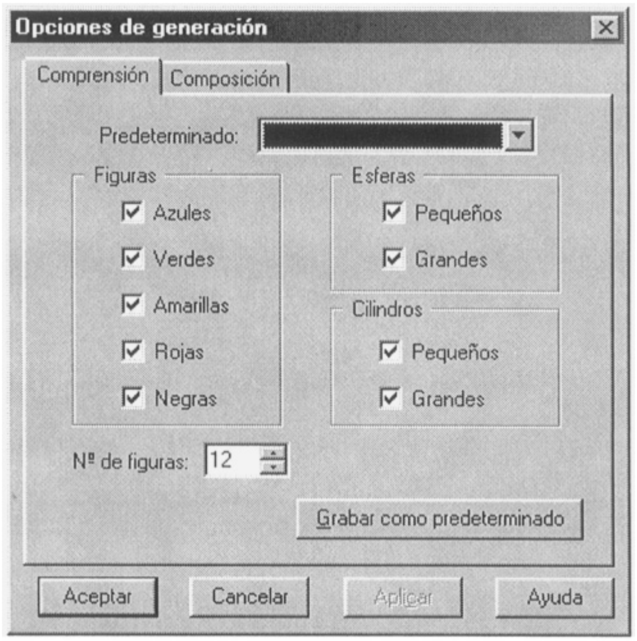
The way these exercises are done is the same as the one with the comprehension exercises, except that at the end of the exercise a representative image of the onomatopoeia will appear.

Table 1.

Onomatopoeia	Sound
Plane	AAA
Donkey	I AAA, I AAA
Horse	III
Pig	o i, o i
Wolf	A UUU, A UUU
Monkey	I I I
Dog	u A u, u A u
Indian	A, A, A
Ghost	UUU, UUU, UUU
Fright	OOO
Scream	EEE, EEE, EEE
Mother	eee aaa, eee aaa, eee aaa
Crying	u AAA, u AAA, u AAA
Hammering	Aiii

4. FUNCTIONALITIES

One of the main characteristics of the application is to generate exercises automatically. This automatic generation can be configured by the teacher to adapt them to the learning needs of the child. There are options that allow configuring these characteristics, as you can see in figure 3, for the comprehension exercises. You can also save personalised configurations, so that you can have configuration templates. In this way, you can use them without having to configure them again.



Predeterminado:

Figuras: ☑ Azules ☑ Verdes ☑ Amarillas ☑ Rojas ☑ Negras

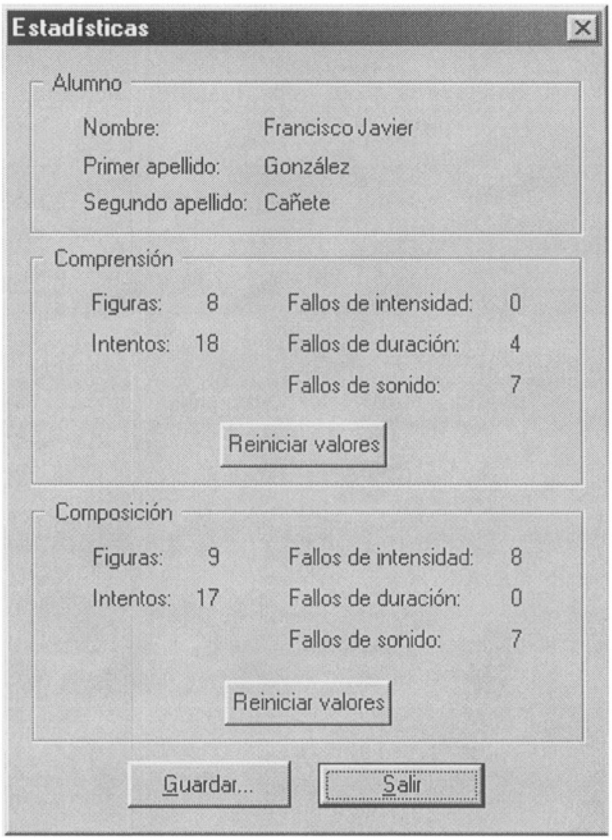
Esferas: ☑ Pequeños ☑ Grandes

Cilindros: ☑ Pequeños ☑ Grandes

Nº de figuras: 12

Figure 3. Comprehension exercises configuration

The pupils' personal data are introduced in the system with a simple form that allows navigating through the files. This allows the identification of the pupils in the system.


Estadísticas

Alumno

Nombre:	Francisco Javier
Primer apellido:	González
Segundo apellido:	Cañete

Comprensión

Figuras:	8	Fallos de intensidad:	0
Intentos:	18	Fallos de duración:	4
		Fallos de sonido:	7

Composición

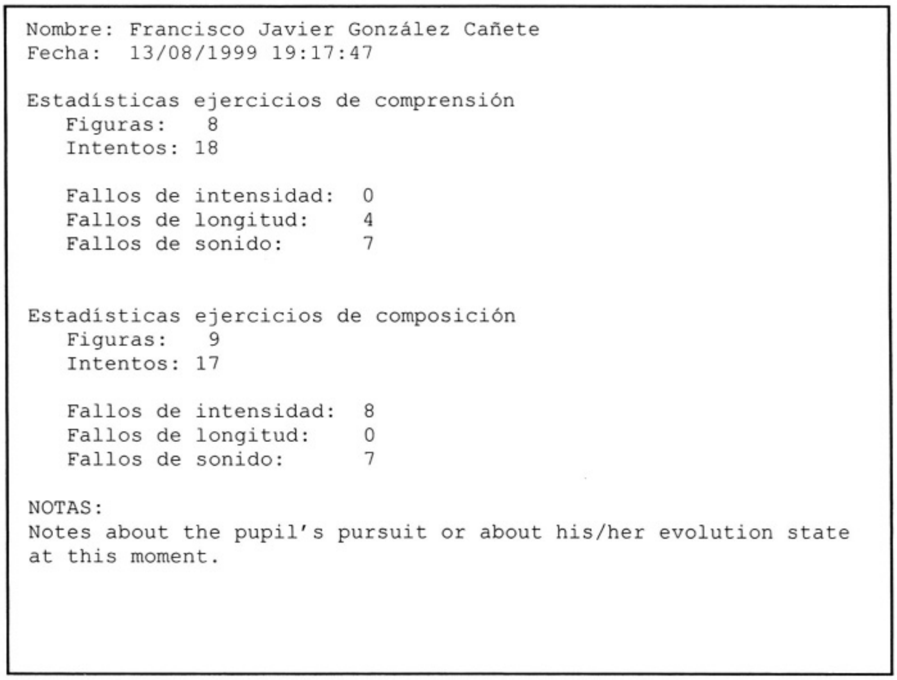
Figuras:	9	Fallos de intensidad:	8
Intentos:	17	Fallos de duración:	0
		Fallos de sonido:	7

Figure 4. Statistics

The system stores personalised statistics for each pupil, in order to allow the teacher to analyse them and make decisions about the exercises that the child will have to do. In this way, the teacher must focus on those aspects that have not obtained the adequate development of the concrete parameter studied, within the preestablished parameters.

The pupil's statistics are shown in figure 4, where:

- **Figuras**: Number of elements of the exercise answered correctly.
- **Intentos**: Number of intents needed to answer the elements indicated before.
- **Fallos de intensidad**: Number of failures in the intensity of the sound to pronounce.
- **Fallos de duración**: Number of failures of the sound duration.
- **Fallo de sonido**: Number of wrong sounds pronounced by the pupil.



```
Nombre: Francisco Javier González Cañete
Fecha:  13/08/1999 19:17:47

Estadísticas ejercicios de comprensión
   Figuras:   8
   Intentos: 18

   Fallos de intensidad:  0
   Fallos de longitud:    4
   Fallos de sonido:      7


Estadísticas ejercicios de composición
   Figuras:   9
   Intentos: 17

   Fallos de intensidad:  8
   Fallos de longitud:    0
   Fallos de sonido:      7

NOTAS:
Notes about the pupil's pursuit or about his/her evolution state
at this moment.
```

Figure 5. Statistics file example

This statistics is stored in files, so that the teacher can control the pupil's evolution. The teacher can add a little comment to the statistics about the pupil's state, evolution, etc... (figure 5).

The final presentation of the exercises on the pupil's screen is highly configurable. Most of the procedures can be configured.

5. CONCLUSIONS

Collares Ortofónicos is, firstly, a multimedia environment for the development of the suprasegmental parameters of speaking, concretely for the intensity and duration or time. The user-friendly aspects, the presentation and the visual and hearing reinforcements have been taken into account. The application has the friendly accessibility characteristics needed to apply to children. It is also an environment with a wide set of applications, it depends on the psycholinguistic or altered cognitive processes, that is, onset phonemes per second. Onset voicing (VOT voice onset time), from left to right eyes movement, pattern learning and discrimination, successive and simultaneous processing, etc. The teacher imposes the application usage restrictions, since it has wide-open possibilities for imagination and creativity.

Among the possible improvements of the actual state of the application, we can propose adding more phonetic capabilities, and not only vocal phonemes; to add syllable capabilities or even whole words.

As a possible development, we could think of substituting the teacher with an intelligent tutor directed by the computer. In this way, the virtual tutor will propose the exercises depending on the pupil's statistics and evolutions for the correct development of the cognitive and linguistic aspects.

REFERENCES

1. Torres, S. (1989). Nuevas tecnologías de la comunicación e información, de orientación oralista, aplicadas a la rehabilitación del deficiente auditivo. La tarjeta de voz de IBM – Speech Viewer. España: Federación Ibérica de Asociaciones de Padres de Sordos (FIAPAS), 11, Separata I - XVI.
2. Das, J.P. (1996). Más allá de una escala unidimensional para la evaluación de la inteligencia. In S. Molina y M. Fandos (coord.), *Educación cognitiva.* Zaragoza: Mira editores. 77-92.
3. Das, J.P. (1988). Simultaneous-successive processing and planning. In R. Schmeck (ed.), *Learning Styles and Learning Strategies.* New York: Plenum.
4. McGregor, G. (1994). Physical Characteristics Assessment (PCA). Computer Acces for Individual with Cerebla Palsy. USA: Don Johnston.
5. Gajic, K. (1996). La rehabilación individual del niño con audición deficiente. Fiapas, 50. I-VIII.
6. Pinker, S. (1994). The language instinct. How the mind creates languages. New York: Morrow. [Translation: El instinto del lenguaje. Madrid: Alianza, 1995].
7. Neville, H.J. (1997). Developmental Specificity in neurocognitive development in human. In M.S. Gazzaniga (ed.), The cognitive neurosciences, 219-231.
8. Mehler, J. and Dupoux, E. (1990). Naître humaine. Paris: Odile Jacob [Translation: Nacer sabiendo. Madrid: Alianza, 1992].
9. Mehler, J., Jupoux, E.,Nazzi, T.and Dehaene-Lambert, G. (1996). Coping with Linguistic Diversity: The Infant's Viewpoint. In Morgan, J. L. and Demuth, K. (eds.). Signal to Syntax: Bootstrapping from Speech to Grammar in Early Language Acquisition. Hillsdale, NJ: Erlbaum.
10. MOC (1992-2000). Materiales para intervención. Documentos del Proyecto Modelo Oral Complementado. Universidad de Málaga.
11. IBM (1997). IBM SpeechViewer III para Windows. Guía de Usuario.
12. Aguilera, S. (1995). Análisis y parametrización de la voz como ayuda a la logopedia. In S. Aguilera (coord.), Nuevas tecnologías aplicadas a la discapacidad. Madrid: Ministerio de Asuntos Sociales.

EXercita
A System for Archiving and Publishing Programming Exercises

C. Gregorio Rodríguez*, L.F. Llana Díaz*, R. Martínez Unanue*, P. Palao Gostanza*, C. Pareja Flores**, and J.Á. Velázquez Iturbide***
**Facultad de Informática,*
Dpto. Sistemas Informáticos y Programación, U. Complutense de Madrid
e-mail: cgregorio@sip.ucm.es, llana@sip.ucm.es, raquel@eucmos.sim.ucm.es, ecceso@eucmos.sim.ucm.es
***Escuela Universitaria de Estadística,*
Dpto. Sistemas Informáticos y Programación, U. Complutense de Madrid
e-mail: cpareja@sip.ucm.es
****Escuela Superior de Ciencias Experimentales y Tecnología,*
Universidad Rey Juan Carlos de Madrid
e-mail: a.velazquez@escet.urjc.es

Key words: Computer science education, programming exercises, document repository, electronic publishing, LaTeX, PostScript, Web pages

Abstract: We present eXercita, a system designed to archive and publish programming exercises. It consists of a repository of structured documents, each one describing a complete exercise (problem statement, solution, history, examples...), and several tools to manage it. Documents are written in the eXercita language, which is an extension of the documental language LaTeX. Programming exercises can be catalogued and searched according to several criteria of interest to the user. Besides, they can be automatically published as PostScript files or Web pages. EXercita can be used by a teacher to elaborate a set of problems for his/her course or by a student for self-evaluation. In a first phase, we have loaded the database with exercises of a first course of programming.

1. INTRODUCTION

This work is an answer to a need perceived by the authors, who have been teaching programming during the last few years, preparing and updating attractive and original exercises (examples, problems, assignments, exams, etc.). Most of these materials can be reused by other teachers, so we needed some practical way to share these resources using information and communication technologies [1].

Benefits of such reusability are remarkable. Teachers can make use of other teachers' experience and materials, which are usually high-quality exercises. Besides, the teacher, has more time, either to design new exercises or to work in other tasks. If the repository is available to the students, they have the stimulus of studying with a collection of original, attractive, wide-ranged, and high-quality material.

The simplest solution was to develop a repository with the documents generated by each teacher. However, this solution is too rigid, because a myriad of different circumstances can recommend changing them slightly. For instance, the exercises can be designed for different purposes (exams, examples for lectures, etc.), they can be used for different courses, the teachers have different preferences in presentation, they might want to deliver different parts to the students (e.g. whether to include hints or a solution), etc. Thus, a more flexible organisation of this material, capable of coping with different uses, was desirable.

We have structured every exercise so that it is composed of different and clearly established pieces that are marked-up to be processed flexibly and automatically. Based on this philosophy, we designed a mark-up language (called eXercita) to write exercise documents. We built a database of exercises, and developed several tools to handle the database, to process documents, and to convert them into printable files and Web pages.

The paper is organised as follows. In the second section, we describe the main requirements for the design of eXercita. Sections 3 and 4 respectively describe the tools that constitute the eXercita system and the main features of the eXercita language. Finally, sections 5 and 6 contain a brief discussion, including related work, and our conclusions and future work.

2. REQUIREMENTS FOR EXERCITA

The main requirements that guided our design decisions for eXercita can be summarised as:

– The user should be able to retrieve exercises in two formats: the source document as a text file or the electronic document in a high-quality format. These choices were made concrete as LaTeX documents for the first requirement and PostScript files and Web pages for the second one. The first choice was adopted seeking flexible handling and maintenance of documents. The second choice was adopted seeking practical use and wide dissemination of exercises.

- The teacher should be able to select exercises according to several criteria by providing the corresponding keywords, such as subject or concepts involved.
- The teacher should be able to control the level of difficulty and length of the exercises. A given exercise can be configured according to the expected difficulty; for instance, a varying number of sections may be included and may contain hints to solve it. Depending on the choice made, both the extension and the difficulty vary.
- The teacher should be able to establish a context for the exercises. For instance, programming exercises that use certain data structures can be used in different environments: combinatorics, matrix algebra, games, etc.
- The students should be able to select solved exercises for self-evaluation. This way, their can solve the exercise and check their solution against the solution proposed in the exercise.
- The student should be able to use cross references among exercises to practise with a family of exercises. Cross references can point to other exercises where a similar question is asked.
- The student should be able to say whether she wants to make visible useful information about the exercise (hints, bibliography, examples, etc.). This function allows the student to face an exercise with different levels of knowledge about involved concepts.

Based on these requirements, we have developed the tools we describe below.

3. TOOLS IN EXERCITA

There are two kind of tools developed for eXercita: the database of source documents and the conversion tools to convert the documents into an electronic publication.

3.1 The database

The database can be updated and queried by a user. Since we expect a large number of problems in our database, it is necessary to provide several cataloguing criteria. Our first idea was cataloguing the database with respect to the schedule of a course, e.g. a first programming course. Therefore, we would find first the exercises about assignment instructions, then those about conditionals, then about iterations and so on. The problem about cataloguing exercises with this criterion is that it restricts the database to a preconceived idea of a programming course, i.e., it is not a general purpose database. As a conclusion, we need cataloguing criteria independent from a particular course structure.

The cataloguing criteria we have finally included are:

– The **subject** of the exercise. A problem about calculating the factorial of a number will be catalogued in mathematics/combinatorial.
– The **difficulty** and the **length** of the exercise. These two criteria are often different. A particular exercise could be very difficult, but it could be solved in a few lines. Or the reciprocal, it could be very easy but requiring several pages.
– The **kind** of the exercise. The exercise can be a laboratory assignment, a class example, an exercise or an exam.
– The **part** of a programming course where the exercise is considered; for example, in iterations or subprograms.

Queries will give as a result one or several exercises. This output can be obtained in two different formats:

– A paper-printable file. The eXercita language is a metalanguage over LaTeX, so that, once the exercises are translated into LaTeX, the output can be a high quality written document.
– An electronic publication. The selected exercises are processed in order to obtain the proper input for the LaTeX2HTML [2, 3] translator. Therefore, the output is an HTML document. We describe here the two tools that we have developed to transform exercises into a readable medium, either a printable document or a Web page.

3.2 Conversion tools

EXercita provides several tools to produce documents. Fig. 1 illustrates them graphically, where *exercita.sty* is a package (written in TEX) that allows embedding eXercita in LaTeX and, therefore, generating PostScript code, whereas ej2HTML translates eXercita into HTML. The eXercita manual explains the use of these tools in full detail. We explain both modes of publication in a more detailed way.

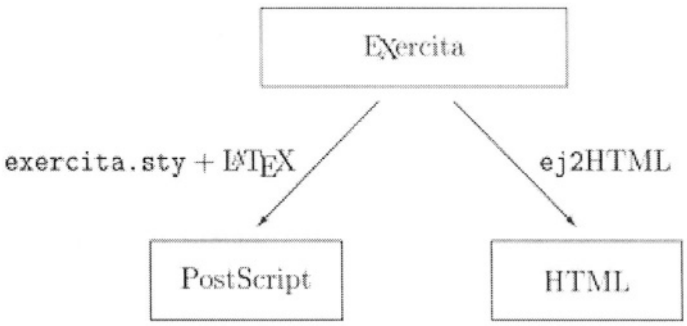


Figure 1. Processing of eXercita documents.

3.2.1 Publishing exercises with LaTeX

The most common use of eXercita is generating a paper document, i.e., a file in a printing format, such as PostScript. We could have built a compiler to transform eXercita code into LaTeX code, so that a printed document could be obtained in a second phase. A better alternative consists in building a set of LaTeX packages that translate directly text written in eXercita into such a format. In this way, we avoid having to process in two or more steps and we gain in portability, since TeX is universal for any computer.

The processing of eXercita with LaTeX is divided into two levels:

– At the lower level we can find an analysis engine for eXercita marks. Although such an engine analyses blindly this marks, it can be configured, indicating what marks must be detected and their corresponding handling. This level can be configured at a low effort, but it requires certain knowledge of LaTeX. This level will be modified only in limited situations.

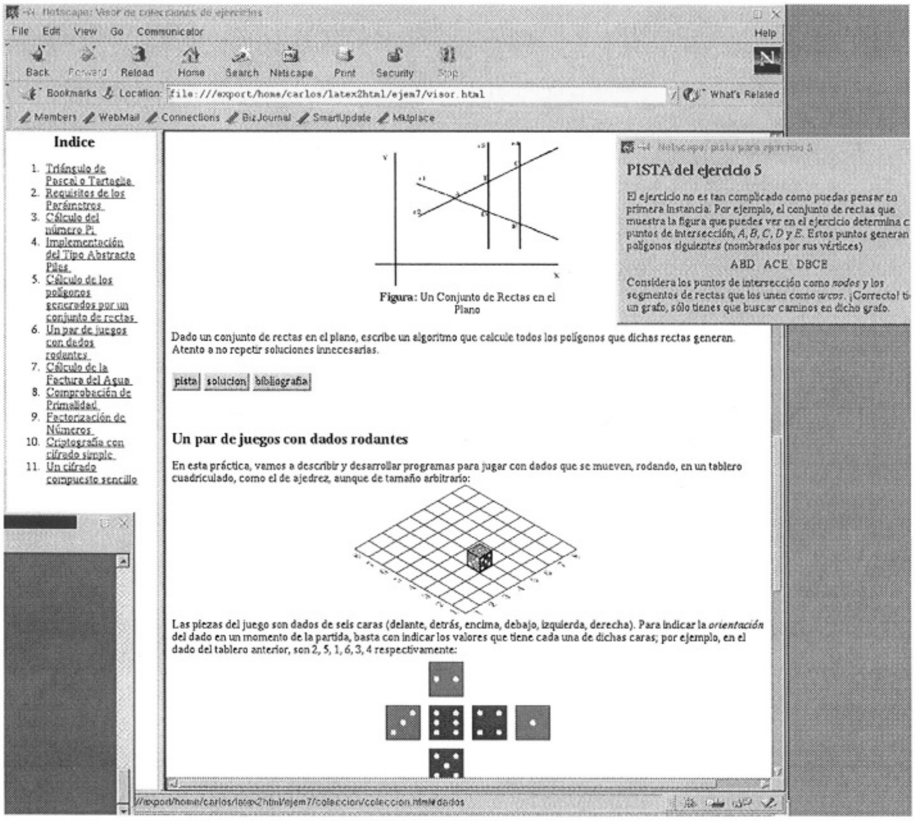
– The upper level can contain different classes of *packages* and *classes* (styles). Currently, we have only developed a class (*uniexer*) that includes parameters that allow configuring it for very different purposes.

– Fig. 2 shows a page printed from a PostScript file generated by eXercita.

3.2.2 Publishing exercises on the Web

The target language for publishing in the net is HTML. We also use JavaScript to make presentations more comfortable, active and communicative. The final appearance can be easily modified thanks to style sheets.

The particular tool used to generate a Web document from a collection of exercises is ej2HTML, which basically consists in the sequential composition of ej2TeX and the well-known LaTeX2HTML.


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: file:///export/home/carlos/latex2html/ejem7/visor.html
What's Related
Members WebMail Connections BizJournal SmartUpdate Marketplace
pista solucion bibliografia
//export/home/carlos/latex2html/ejem7/coleccion/coleccion.html#dados

Indice

1. Triángulo de Pascal o Tartaglia.
2. Requisitos de los Parámetros.
3. Cálculo del número Pi.
4. Implementación del Tipo Abstracto Pilas.
5. Cálculo de los polígonos generados por un conjunto de rectas.
6. Un par de juegos con dados rodantes.
7. Cálculo de la Factura del Agua.
8. Comprobación de Primalidad.
9. Factorización de Números.
10. Criptografía con cifrado simple.
11. Un cifrado compuesto sencillo.

Figura: Un Conjunto de Rectas en el Plano

Dado un conjunto de rectas en el plano, escribe un algoritmo que calcule todos los polígonos que dichas rectas generan. Atento a no repetir soluciones innecesarias.

PISTA del ejercicio 5

El ejercicio no es tan complicado como puedas pensar en primera instancia. Por ejemplo, el conjunto de rectas que muestra la figura que puedes ver en el ejercicio determina c puntos de intersección, *A, B, C, D* y *E*. Estos puntos generan polígonos siguientes (nombrados por sus vértices)

ABD ACE DBCE

Considera los puntos de intersección como *nodos* y los segmentos de rectas que los unen como *arcos*. ¡Correcto! ti un grafo, sólo tienes que buscar caminos en dicho grafo.

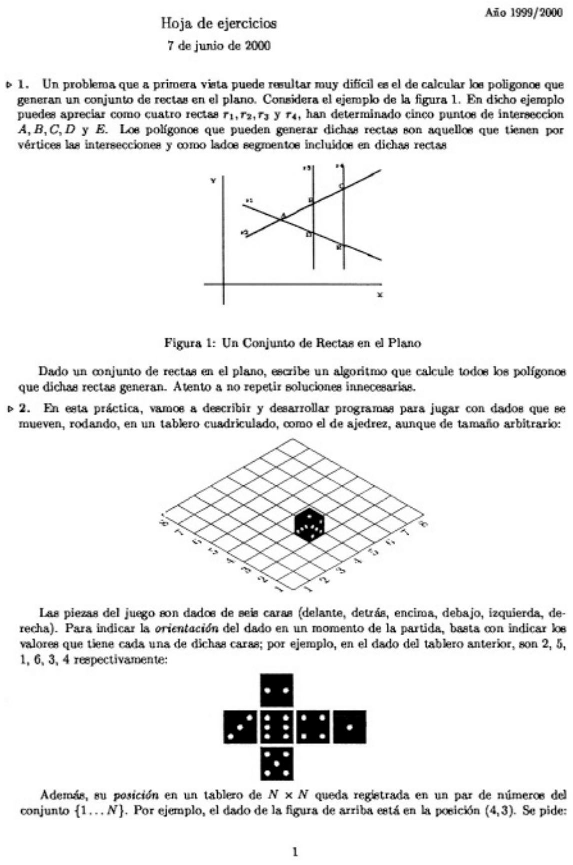
Un par de juegos con dados rodantes

En esta práctica, vamos a describir y desarrollar programas para jugar con dados que se mueven, rodando, en un tablero cuadriculado, como el de ajedrez, aunque de tamaño arbitrario:

Las piezas del juego son dados de seis caras (delante, detrás, encima, debajo, izquierda, derecha). Para indicar la *orientación* del dado en un momento de la partida, basta con indicar los valores que tiene cada una de dichas caras; por ejemplo, en el dado del tablero anterior, son 2, 5, 1, 6, 3, 4 respectivamente:

Figure 2. A sample of Web page generated by eXercita

The structure of the generated HTML file is based on encapsulating information in layers that can be handled with the JavaScript language and allows presenting the description of exercises in a practical and dynamic way, showing the less relevant information only when the user requires it.



Año 1999/2000

Hoja de ejercicios

7 de junio de 2000

▹ **1.** Un problema que a primera vista puede resultar muy difícil es el de calcular los poligonos que generan un conjunto de rectas en el plano. Considera el ejemplo de la figura 1. En dicho ejemplo puedes apreciar como cuatro rectas r_1, r_2, r_3 y r_4, han determinado cinco puntos de interseccion A, B, C, D y E. Los polígonos que pueden generar dichas rectas son aquellos que tienen por vértices las intersecciones y como lados segmentos incluidos en dichas rectas

Figura 1: Un Conjunto de Rectas en el Plano

Dado un conjunto de rectas en el plano, escribe un algoritmo que calcule todos los polígonos que dichas rectas generan. Atento a no repetir soluciones innecesarias.

▹ **2.** En esta práctica, vamos a describir y desarrollar programas para jugar con dados que se mueven, rodando, en un tablero cuadriculado, como el de ajedrez, aunque de tamaño arbitrario:

Las piezas del juego son dados de seis caras (delante, detrás, encima, debajo, izquierda, derecha). Para indicar la *orientación* del dado en un momento de la partida, basta con indicar los valores que tiene cada una de dichas caras; por ejemplo, en el dado del tablero anterior, son 2, 5, 1, 6, 3, 4 respectivamente:

Además, su *posición* en un tablero de $N \times N$ queda registrada en un par de números del conjunto $\{1 \ldots N\}$. Por ejemplo, el dado de la figura de arriba está en la posición $(4, 3)$. Se pide:

1

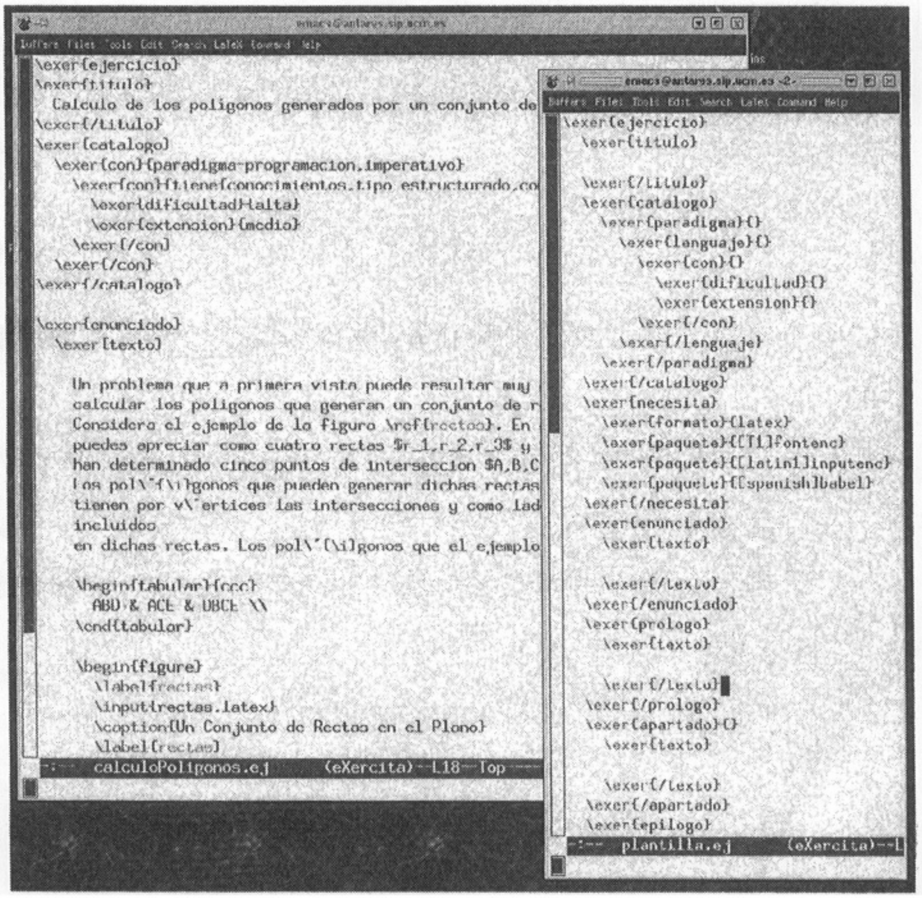
Figure 3. A sample of PostScript document generated by eXercita

The appearance of the generated document keeps the quality of LaTeX, including images of mathematical formulas, figures, etc.

We have also provided several improvements for browsing over a plain Web page. The same window shows the index of exercises generated by ej2TeX in order to facilitate navigation. In addition, several parts of the exercise can be shown on demand, in which case a new window is opened.

4. THE EXERCITA'S MARK-UP LANGUAGE

The eXercita language is a mark-up language designed to represent, store and manage exercises in a database. EXercita is just a layer over LaTeX that is oriented to a specific domain, programming exercises. Any exercise will be written with LaTeX and marked-up with eXercita.



```
\exer{ejercicio}
\exer{titulo}
  Calculo de los poligonos generados por un conjunto de
\exer{/titulo}
\exer{catalogo}
  \exer{con}{paradigma-programacion.imperativo}
    \exer{con}{tiene{conocimientos.tipo estructurado.co
      \exer{dificultad}{alta}
      \exer{extension}{medio}
    \exer{/con}
  \exer{/con}
\exer{/catalogo}

\exer{enunciado}
  \exer{texto}

    Un problema que a primera vista puede resultar muy
    calcular los poligonos que generan un conjunto de r
    Considera el ejemplo de la figura \ref{rectas}. En
    puedes apreciar como cuatro rectas $r_1,r_2,r_3$ y
    han determinado cinco puntos de interseccion $A,B,C
    Los pol\'{\i}gonos que pueden generar dichas rectas
    tienen por v\'ertices las intersecciones y como lad
    incluidos
    en dichas rectas. Los pol\'{\i}gonos que el ejemplo

    \begin{tabular}{ccc}
      ABD & ACE & DBCE \\
    \end{tabular}

    \begin{figure}
      \label{rectas}
      \input{rectas.latex}
      \caption{Un Conjunto de Rectas en el Plano}
      \label{rectas}
-:--  calculoPoligonos.ej      (eXercita)--L18--Top
```

```
\exer{ejercicio}
  \exer{titulo}

  \exer{/titulo}
  \exer{catalogo}
    \exer{paradigma}{}
      \exer{lenguaje}{}
        \exer{con}{}
          \exer{dificultad}{}
          \exer{extension}{}
        \exer{/con}
      \exer{/lenguaje}
    \exer{/paradigma}
  \exer{/catalogo}
  \exer{necesita}
    \exer{formato}{latex}
    \exer{paquete}{[T1]fontenc}
    \exer{paquete}{[latin1]inputenc}
    \exer{paquete}{[spanish]babel}
  \exer{/necesita}
  \exer{enunciado}
    \exer{texto}

    \exer{/texto}
  \exer{/enunciado}
  \exer{prologo}
    \exer{texto}

    \exer{/texto}
  \exer{/prologo}
  \exer{apartado}{}
    \exer{texto}

    \exer{/texto}
  \exer{/apartado}
  \exer{epilogo}
-:--  plantilla.ej      (eXercita)--L
```

Figure 4. A sample of eXercita's mark-up language

The full description of eXercita language is beyond the space limits of this paper. Thus, we simply give a broad classification of the marks in order to give an idea of its expressive power:

- **Customisation parameters.** The teacher can specify date (with different formats), subject, course, group, academic year, institute, etc.

- **Parts of the exercise.** We can declare title, purpose, abstract, text, prologue, enunciation, section, solution, hint, example, epilogue, history, bibliography, etc.
- **Other marks.** They allow including logos, parameters, cross references to other exercises, conditional processes, etc.
- **Requirements.** They deal with dependencies among exercises, LaTeX styles and packages, etc.
- **Cataloguing.** Index keys are concepts involved, estimated difficulty level, size, etc.

Notice also that all the categories are general enough to be used for any subject matter. Our orientation to programming exercises is only constrained by including several specific cataloguing marks like programming paradigm, target programming language or data structures involved.

Fig. 4 gives a glimpse of the work with eXercita. The left window in the figure shows a part of the first exercise contained in Figs. 2 and 3, as written in the eXercita language. The right window shows a template automatically generated by Emacs in eXercita mode to edit exercises.

A full description of the eXercita language can be found in the user's manual [4].

5. Discussion and Related Work

There are alternatives to LaTeX for the mark-up of exercises, being the best alternatives, in our opinion, HTML and XML. The rest of the languages (PDF, RTF) either are not readable, or are only oriented to the automatic generation and treatment of documents, or do not allow managing the structure of documents, but only their presentation. We have chosen LaTeX instead of HTML because the former is more concise, it is more usually generated by hand, it is oriented to produce high quality printed documents, there are good LaTeX-to-HTML translators, and we have a large experience using LaTeX. XML is also a good alternative that we plan to consider in the future.

There is not much work related to eXercita. In 1997, Deborah Knox proposed the development by ACM SIGCSE of a repository of computer science laboratories [5, 6]. They should be high-quality laboratory materials to be shared by the community of CS educators. A first prototype included a database of labs information (included links to full materials) accessed by means of the Web, and related facilities for lab submission, searching by keyword or topic, and an annotation capability for each lab. Currently, *The Lab Repository* is one of the technical sections of the *SIGCSE Bulletin.* In its current state [6], it is peer reviewed to ensure high quality and there are five labs available.

Another effort by ACM SIGCSE to share teaching resources on the Web is *Computer Science Education Links* [7]. It is a Web page maintained by Renée McCauley with links to different kinds of materials to teach computer science. It is structured by categories (currently, there are 16 categories with a total of 119 links). There is also a technical section in the *SIGCSE Bulletin* for this effort. Similar resources for teaching object-oriented programming in the first course can be found at [8].

The system most similar to eXercita is SAIL [9]. It is also based on LaTeX and a database of exercises. However, the emphasis and its main features are quite different. It allows generating different instances of the same problem in order to encourage cooperation among students without fear of plagiarism. SAIL also provides support for grading. However, it provides a much less comprehensive support and structuring for exercises. Therefore, its searching and publishing capabilities are more restrictive, since it simply provides a facility for searching based on text matching.

6. CONCLUSIONS AND FUTURE WORK

We have designed eXercita, a very expressive mark-up language that allows creating databases of exercises as well as their recovery by several users. We have also developed tools to convert exercises from the database into a PostScript file or into an HTML page.

Using eXercita requires a previous effort consisting in annotating exercises with marks, especially cataloguing marks. Although it may seem that cataloguing an exercise requires a great effort, it is a task that a teacher does implicitly when thinking of a problem for a given topic or when seeking one in a bibliographical search. Ideally, this effort will be made once, fostering reuse and improving the teacher's performance. We are currently collecting programming exercises and annotating them with the eXercita language; some of them are accessible at [10].

Another aspect we want to remark is the ease of extension of the database and associated tools to any discipline requiring practice with exercises for learning. We are loading the database with exercises from a first course on programming since it is a subject where all the authors have experience. However, representing and storing exercises for another course would require an adaptation effort consisting of:

- Identify parameters associated to the subject matter.
- Design cataloguing marks.

In the future, we plan to complete and improve the interface with the database. We also plan to make public our database to teachers and students of computer programming, within our universities and also in others. In addition, we must establish some criteria in order to use it adequately. We also expect that feedback from users will help us in fine-tuning such criteria. Finally, we are considering using eXercita for other subject matters, such as Mathematics or Physics.

ACKNOWLEGMENTS

This work was supported by the Universidad Complutense de Madrid under the educational innovation project *Elaboración de material educativo para prácticas de programación,* ref. PIE 99/32.

REFERENCES

1. R. Jiménez Peris, M. Patiño Martínez, C. Pareja Flores and J.Á. Velázquez Iturbide, "New technologies in computer science education", in *Computer Science Education in the 21st Century,* Tony Greening (ed.), Springer-Verlag, 2000, pp. 113-136
2. N. Drakos, *The LATEX2HTML Translator,* available at http://www.desy.de/asgdocs/latex2html/manual-99.1/index.html, March 1996
3. N. Drakos, LateX2HTML Development Repository, available at http://saftsack.fs.uni-bayreuth.de/~latex2ht/, November 1999
4. C. Gregorio Rodríguez et al., *EXercita: manual de uso,* technical report SIP 102-00, Dept. Sistemas Informáticos y Programación, Universidad Complutense de Madrid, May 2000
5. D. Knox, "On-line publication of CS laboratories", *Proceedings of the Twenty-Eighth SIGCSE Technical Symposium on Computer Science Education,* in *SIGCSE Bulletin,* vol. 29, n. 1, March 1997, pp. 340-344
6. D. Knox, *The Computer Science Teaching Center,* available at http://www.cstc.org/, July 1999
7. R. McCauley, *Computer Science Education Links,* available at http://www.cacs.usl.edu/~mccauley/edlinks/, July 1999
8. J. Bergin, V.K. Proulx et al., "Resources for next generation introductory CS courses", *SIGCSE Bulletin,* vol. 31, n. 4, December 1999, pp. 101-105
9. S. Kovourov et al., "SAIL: A system for generating, archiving and retrieving specialized assignments using LaTeX", *Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education* in *SIGCSE Bulletin,* vol. 32, n. 1, March 2000, pp. 300-304
10. C. Gregorio Rodríguez et al., *EXercita,* http://aljibe.sip.ucm.es, October 2000

Using Simulation and Virtual Reality for Distance Education

Juan de Lara, Manuel Alfonseca
Dept. Ingeniería Informática, Universidad Autónoma de MadridCampus de Cantoblanco, 28049 MadridE-Mail: {Juan.Lara, Manuel.Alfonseca}@ii.uam.es

Key words: Continuous Simulation, Web-based simulation, Virtual Reality, Distance Education.

Abstract: This paper describes the construction of virtual reality simulations for distance education through the Internet. This is accomplished by means of an object oriented continuous simulation language, called *OOCSMP*, and a Java generating compiler for this language called *C-OOL*. This compiler is also able to create VRML worlds. The behaviour of the VRML world is specified in the *OOCSMP* models. Change of simulation parameters is possible at run time by means of a Java interface, generated by the compiler. An example of the simulation of the inner Solar System is presented.

1. INTRODUCTION

The growing popularity of the Internet, and the increasing number of computers connected to it, makes it an ideal framework for distant education. Not only educational sciences, but also a large number of disciplines are re-thinking their traditional philosophies and techniques to adapt to the new technologies [1]. One of these disciplines is computer simulation.

Virtual Reality (VR) techniques offer immersive environments in which the user has great possibilities of interaction. The emergence of the Virtual Reality Modelling Language VRML [2] and browser plug-ins for this language [3] has made it possible to build virtual worlds accessible through the Internet. The VRML97 [4] standard defines an external API that allows us to control VRML objects from a Java applet.

Its advantages, such as new possibilities of interaction and more realistic and pleasant learning environments, are turning VR into a valuable tool in distance education [5,6].

In this paper, we present some improvements to the tools that we use to automatically generate web courses based on simulation [7]. These enhancements include the possibility of incorporating VR panels to the simulation applets. An example with the simulation of the Inner Solar System is presented.

2. OUR SIMULATION TOOLS

The *OOCSMP* continuous simulation language was conceived in 1997 [8] as an object oriented language. The language is specially suited when the system can be modelled with similar interacting objects. A compiler (C-*OOL*) was built for this language in order to produce C++ code or Java applets with the simulation models. This approach simplifies the generation of simulation based web courses. Several web courses have been generated using this language: courses on gravitation, partial differential equations, ecology and basic electronics. These can be accessed at:

`http://www.ii.uam.es/~jlara/investigacion`

The language and the compiler have been designed with an educational focus, for example:

- It is possible to include several forms of output displays in the same simulation.
- Multimedia elements can be synchronised with the simulation execution.
- The user interface allows changing parameters, object attributes or even adding or deleting objects during the simulation execution.
- Alternative simulations can be designed and can be accessed from the main simulation. In this way, the teacher can identify interesting situations that arise when changing some parameter, or when adding some object, etc.
- There are also instructions that allow us to describe the appearance of the web page where the simulation models are going to be placed.

3. EXTENDING THE LANGUAGE TO HANDLE VR

In *OOCSMP,* there is the possibility of associating an icon to each object taking part in the simulation. This icon can be used in all the output forms. The VR extensions allow us to assign also a *VRML* node to each object, by means of the new attribute type `VRMLobject`

A new instruction (`VRMLworld`), to include *OOCSMP* objects (considered as dynamic components of the Virtual World) in a virtual world, has also been added. This instruction also configures the virtual world with static elements.

For example, suppose we want to simulate the behaviour of the inner solar system [9]. To model it, we will encapsulate the behaviour of a planet in a class, and declare an object associated with each planet of the solar system. The class has an attribute to assign a VRML object to the planet. A scheme of the *OOCSMP Planet* class is shown in listing 1.

```
CLASS Planet {          * Definition of Planet class
  NAME          name    * Name of the planet
  VRMLobject    obj     * VRML object associated with the Planet
  DATA M, X0, Y0, XP0, YP0, FI
 INITIAL      * Compute initial data (FIR, CFI and SFI)
  FIR:=FI*PI/180
  CFI:=COS(FIR)
  SFI:=SIN(FIR)
 DYNAMIC   * Distance to the Sun
  R2    := X*X+Y*Y
  R     := SQRT(R2)
  ...
ACTION Planet P   * Mutual actions of two planets
  DPP2 := (P.X-X)*(P.X-X)+(P.Y-Y)*(P.Y-Y)+(P.Z-Z)*(P.Z-Z)
  DPP  := SQRT(DPP2)
  ...
FINISH R<.05
VRMLworld X, Y, Z}
```

Listing 1. Outline of an *OOCSMP* class representing a planet.

The previous *OOCSMP* class defines some blocks (`DYNAMIC` and `ACTION`) to simulate the behaviour of a particular planet, whose full listing can be found at [7]. It also declares a termination condition (`FINISH`), and inserts all the objects of the *Planet* class in the virtual world (`VRMLworld`), where they will be the dynamic elements. The three parameters (`X, Y, Z`) control the position of the object inside the virtual world.

Listing 2 shows the remainder of the *OOCSMP* model, which uses the *Planet* class to generate a model of the inner Solar System.

```
[1] * Universal data, and Sun data
[2]  DATA G:=0.00011869, PI:=3.141592653589793, MS:=332999
[3]  INCLUDE "Planet.csm"
[4]  Planet Mercury("Mercu", “mercury.wrl”, 0.055271,       ...)
[5]  Planet Venus  ("Venus", “venus.wrl”, 0.81476,  0.7233,...)
[6]  Planet Earth  ("Earth", ”earth.wrl”, 1, 0,       1,    ...)
[7]  Planet Moon   ("Moon", “moon.wrl”, 0.01235,  0,        ...)
[8]  Planet Mars   ("Mars", “mars.wrl”, 0.10734,  1.5233,   ...)
[9]  Planet Apollo ("Apolo", “Apollo.wrl”, 1957E-14,        ...)
[10] Planet Jupiter("Jupit", “jupiter.wrl”, 317.94,         ...)
[11] Planet InnerSystem := Mercury, Venus, Earth, Moon, Mars,
[12]                       Apollo, Jupiter
[13] DYNAMIC
[14]   InnerSystem.STEP()
[15]   InnerSystem.ACTION(InnerSystem)
[16] TIMER delta:=.0005, FINTIM:=2, PRdelta:=.1, PLdelta:=.01
[17] VRMLworld “sun.wrl”
[18] METHOD ADAMS
```

Listing 2. The OOCSMP model for the inner solar system.

Line 17 adds the sun (a static element) to the VRML world. When declaring each planet, we have to specify the VRML node associated with it. In our case, each VRML node will be a sphere with a hyperlink. When the user clicks in the hyperlink, useful data of the planet will be presented in an HTML frame. For example, the following listing shows the *wrl* file associated to planet Mars.

```
#VRML V2.0 utf8

Anchor {
    url "mars.html" parameter [ "target=Details" ]
    children [
        Shape {
            appearance  Appearance {
                material  Material{
                   diffuseColor 1 0 0
                   shininess .5
                }
            }
            geometry Sphere { radius 0.25 }
        }
    ]
}
```

Listing 3. VRML node associated with planet Mars.

4. GENERATING A VR SIMULATION

When compiling the model with C-OOL, several files are generated:

– A *wrl* file, containing all the VRML objects declared in the model.
– A Java applet, with the simulation logic, the simulation controls, and the other graphical output forms, if any has been selected in the model. The Java applet allows the user to start and stop the simulation, to change the model parameters, etc. It communicates with the Virtual World by means of the External Authoring Interface (EAI) [10]. Using proprietary technologies such as Netscape's LiveConnect [11] is widely extended [2, 5], but we think our solution is better, because additional libraries are not needed, we are compatible with Internet Explorer, and the communication between the Java applet and the Virtual World is done directly from the program, rather than indirectly through JavaScript.

Figure 1 shows the working scheme of all these components. There is a graphical and numerical Java library placed in the Server (for developing purposes, it can be placed in the client), used by the generated Java programs. When the user accesses the simulation page, the VRML code, the Java applets, and the necessary Java classes are first downloaded from the server, but are then executed locally.

The necessary elements to run the simulations are thus:

– A graphic browser.
– A VRML97-compliant plug-in, such as CosmoPlayer.
– A Java Virtual Machine plug-in, understanding at least Java 1.1 code.

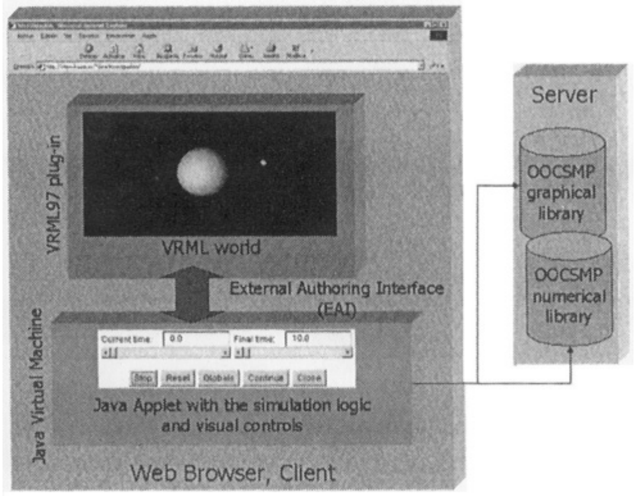
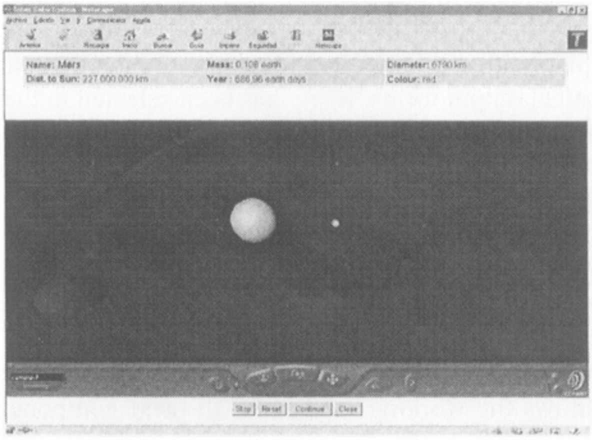


Figure 1. Working scheme of a VR simulation page.

The next figure shows a moment in the execution of the simulation.



Name: Mars | Mass: 0.108 earth | Diameter: 6790 km
Dist. to Sun: 227 000 000 km | Year : 686,96 earth days | Colour: red

Figure 2. A moment in the simulation of the inner solar system.

In the previous figure, the user has clicked on the planet Mars, and the associated data are being shown in the upper frame. The sizes of the planet shapes are not proportional to their real sizes, to make them visible in the image scale. Figure 3 shows a similar simulation, where a two-dimensional plot has been selected as a graphical output form. The animation of both panels is synchronised, because the simulation engine sends messages to CosmoPlayer and to the 2D Java panel every time step.

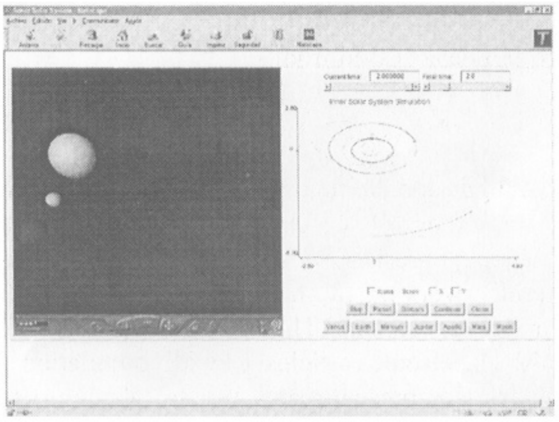


Figure 3: Simulation using Virtual Reality and 2D plots.

There is a problem in the integration of VRML panels in the user interface generated by our compiler, because these panels are not placed inside the Java applet, but in the HTML page in an <*EMBED*> HTML tag. This complicates the construction of the user interface, that we have solved using tables. In figure 3, the VRML world and the applet have both been placed inside an HTML table.

5. CONCLUSIONS AND FUTURE WORK

In the present article, we have presented some extensions to facilitate the construction of Virtual Reality simulations through the Internet. The simulation behaviour is modelled by means of the *OOCSMP* language, which allows us to assign VRML nodes to the *OOCSMP* objects. The compiler takes charge of generating the VR world and the necessary Java files. The programmer only has to model the system behaviour, in a high-level simulation language, and specify the appearance of each object by means of simple VRML nodes. The inclusion of these simulations in educational courses is thus straightforward. In the generated Java code, the inclusion of a VRML panel only represents a few lines of code: some of them in the initialisation, to get a handle to the VRML browser, the others in the main simulation loop (in the function that updates all the graphical output forms), to update the position of the corresponding VRML nodes.

VRML also offers other possibilities of interaction, which we have exploited in the example, such as setting hyperlinks in the simulation objects, which can be used to explain the role of the object in the simulation or, as in our case, to show additional data.

The example can be accessed from:

`http://www.ii.uam.es/~jlara/investigacion/ecomm/solar3.html`

The primitives that add Virtual Reality capabilities to our system only deal with the displacement of objects. Other primitives could also be added to handle other object properties, such as rotation, size, colour, etc.

We are currently working in an authoring tool for the construction of simulated-based web documents. These documents can be educational web courses, but also electronic articles - with simulation and interactive elements - or interactive presentations. We are planning to incorporate the possibility of designing VRML worlds to this tool, or at least allowing an easy interaction with some VRML development tool.

ACKNOWLEDGEMENTS

This paper has been sponsored by the Spanish Interdepartmental Commission of Science and Technology (CICYT), project number TEL 1999-0181.

REFERENCES

1. Page E.H. Buss, A., Fishwick, P.A., Healy, K., Nance, R.E., Paul, R.J. 2000. *Web-Based Simulation: Revolution or Evolution?,* to appear in ACM Transactions on Modeling and Computer Simulation.
2. Hartman, J., Wernecke, J. 1996. *"The VRML 2.0 Handbook. Building Moving Worlds on the Web".* Addison-Wesley.
3. Cosmo Player, at: http://www.cosmosoftware.com/products/player
4. VRML97 Specification: www.web3d.or/Specifications/VRML97
5. Schmid, Ch. 1999. *"A Remote Laboratory Using Virtual Reality on the Web".* Simulation. Special Issue: Web-Based Simulation. Vol.73:1, 13-21.
6. Virtual Reality for Education Resources on the Net: http://www.hitl.washington.edu/projects/knowledge_base/education.html
7. Alfonseca, M., de Lara, J., Pulido, E., 1999. *"Semiautomatic Generation of Web Courses by Means of an Object-Oriented Simulation Language",* Simulation. Special Issue: Web-Based Simulation, Vol 73:1, 5-12.
8. Alfonseca, M., Pulido, E., Orosco, R., de Lara, J. 1997. *"OOCSMP: an object-oriented simulation language".* ESS'97, Passau, pp. 44-48.
9. Alfonseca,M., de Lara, J., Pulido, E. 1998. *"Semiautomatic Generation of Educational Courses in the Internet by Means of an Object-Oriented Continuous Simulation Language".* ESM'98, Manchester, pp. 547-551.
10. Introduction to the VRML and JavaEAI: http://www.dcs.gls.uk/people/personal/snowdonp/vrml/intr.html
11 LiveConnect, at Netscape ftp site: ftp.netscape.com in pub/sdk/plugin/windows/oct_21_97

An Experience on Virtual Teaching: Aula Net

A.J. López Menéndez, R. Pérez Suárez
Universidad de Oviedo

Key words: AulaNet, Virtual Campus, e-learning, Internet, teaching tools

Abstract: The development of new Information and Communication Technologies (ICT) has introduced substantial social changes, including teaching and learning methods. In this context, several projects have been encouraged by European and national authorities, involving both public and private institutions, mainly universities. AulaNet has been developed as a Virtual Campus in the University of Oviedo, offering the subject "Economic Data Analysis" as a first web-based teaching experience. In this work the main aspects of AulaNet are described, including virtual teaching tools, some methodological considerations and a first e-Learning and Teaching balance.

1. AULANET AS A VIRTUAL CAMPUS

Teaching and learning methods have experienced substantial changes during the last years. Printed materials have progressively been complemented by audios, videos and computers. More recently, the introduction of digital technology has opened new possibilities for both the learning process and its technical support.

Aulanet[1] has been developed as a Virtual Campus, trying to make an efficient use of internet. Thus, as shown in figure 1, the website www.aulanet.uniovi.es has been designed as a student-based learning space.

[1] AulaNet is directed by Rigoberto Pérez, coordinator of a teaching team (Ana Jesús López, Covadonga Caso, María Jesús Río and Carmen Ramos) and a computing team (Alberto Fernández, Ignacio Suárez, Juan Carlos Castrillón, Fernando Milla and Marcelino García).

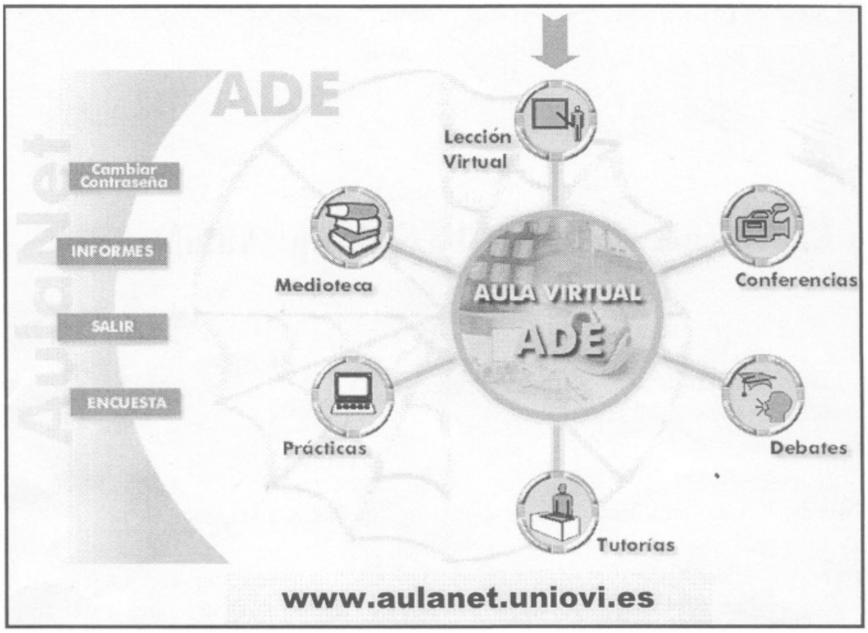


Figure 1. AulaNet Website

This Virtual Campus provides a wide range of teaching tools, including virtual lessons, electronic books, software and presentations. Areas for debates, tutorials, assessment, conferences, ... are easily available from the web.

Each Virtual Lesson starts with a Working Guide, describing its main contents both compulsory (in red) and optional (black). A first approach to each item is provided by a 12-minute presentation[2], complemented by further material as electronic books, animations, interactive questions, practical issues and references. The inclusion of many links, figures and graphs allows the students participation in a flexible, interactive learning process[3].

Once the lesson is finished, a self-assessment is suggested to the students. This option provides a set of test questions randomly selected and automatically corrected through the web.

The practical material is located in a specific area, allowing the selection of some exercises and questions for further submission and evaluation.

[2] Due to the diversity of Internet accessibility conditions, these presentations are available both in audio-visual and only-audio versions.

[3] Electronic books are also available from the Library Area, and documents can be downloaded both in word and pdf format. Nevertheless, this option leads to a static, non-interactive use of the teaching material.

Tutorials are provided in different ways, including on-line assistance, a shared board, chats and video-conferences.

On line assistance connects students with tutors, providing quick answers (within 24 hours) to questions and comments sent by e-mail.

This option has been widely used by the students, complemented by the shared board, which provides an active forum for the debate. Both tools are based in asynchronous communication, since they do not require the simultaneous participation of students.

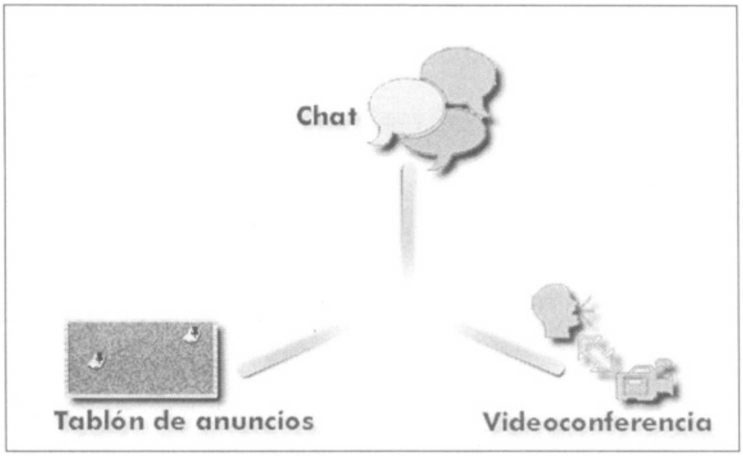


Figure 2. Tutorials in AulaNet

In order to allow synchronous communication, AulaNet also includes chat sessions (periodically held in order to debate about several matters) and video-conferences (providing a face-to-face dialogue between students and teachers).

On-line evaluation is an outstanding characteristic of AulaNet and a distinctive feature from another virtual experiences.

A traditional exam with rigid spatial and temporal restrictions was avoided, since it would introduce incoherencies with the proposed teaching and learning methods. Thus, an on-line assessment was designed allowing a more flexible evaluation process.

According to this method, students apply for a specific examination day (within a previously reserved week), time (five sessions a day) and location (two different campuses). Once the official schedule is published on the web, on-line assessment is held providing each student with a random selected exam.

In order to achieve our objectives, special attention has been paid to the exam design, students' isolation, tutors accessibility and web-cam assisted vigilance.

After the comparison of AulaNet with similar experiences, we point out some distinctive aspects:

- AulaNet has been conceived as a virtual campus in a wide sense, thus including virtual teaching and also web-based academic management.
- The learning and teaching process in AulaNet is fully web-based, removing the use of printed and/or analogical materials.
- Attention should be paid to the differences in the access conditions. Therefore several teaching tools are provided to the students, including video and only-audio versions, electronic mail, chat sessions, video-conferences, ...
- In order to achieve a coherent assessment system, continuous evaluation is complemented with the realisation of a final on-line exam.

2. A FIRST EXPERIENCE WITH "ECONOMIC DATA ANALYSIS"

AulaNet started its activities in February 2000 with the subject "Economic Data Analysis" offered as a free selection subject in the so-called G7 Group[4].

This first subject was specifically designed for virtual teaching and learning, focusing on the student as a user of statistical information, mainly economic data.

As shown in Figure 3, this subject has been followed by 74 students of different academic profiles, with preeminence of computing and economic degrees.

[4] This Group includes Universidad de Cantabria, Universidad de Oviedo, Universidad del País Vasco, Universidad de La Rioja, Universidad de Zaragoza, Universidad Pública de Navarra and Universidad de las Islas Baleares. All these institutions share virtual teaching projects with full academic recognition of the credits.

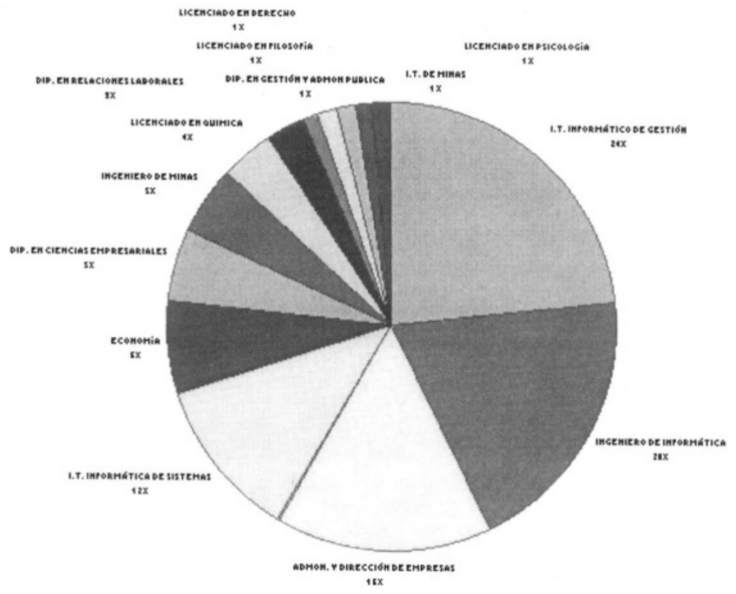
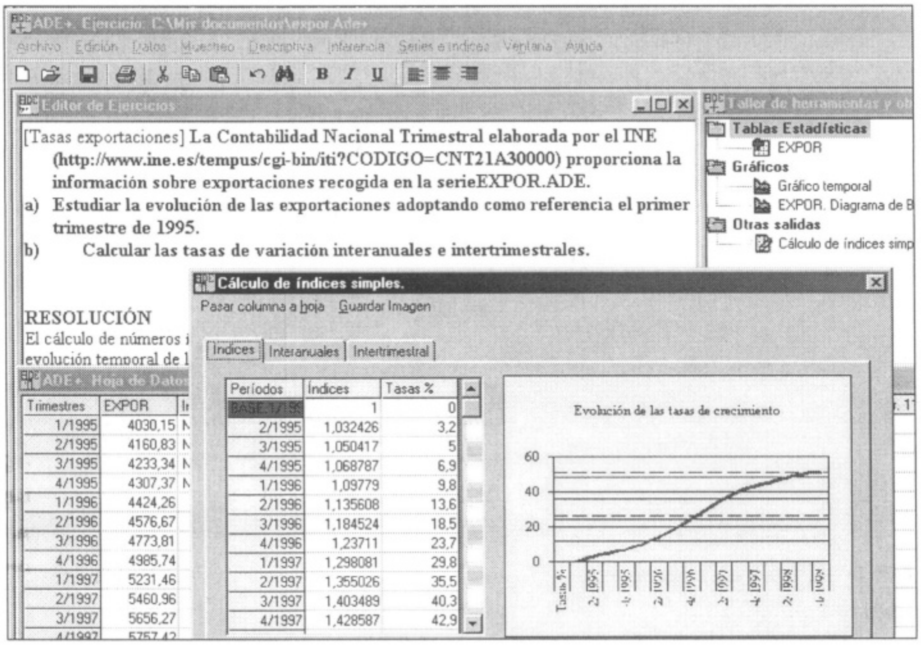


Figure 3. Students of "Economic Data Analysis"

According to this first experience, the interest of the virtual lessons as e-learning tools must be emphasised. More specifically, "Economic Data Analysis" contents have been enriched with links to Statistical Institutes, International Organisations, Economic Reports,... also including a "Statistical Mistakes" section.

The practical material is based on the software ADE+, specifically developed for teaching purposes and available from the web[5]. Students have free access to solved and proposed exercises, and are asked to mail some personal work to the tutor.

[5] The software ADE+ (Análisis de Datos Estadísticos) has been developed and registered by Rigoberto Pérez and Ana Jesús López (license 1866 of the Intelectual Property Provincial Register of Asturias). A first "demo" of this software was also distributed with the book Pérez, R.; López, A.J. (1997): *Análisis de datos económicos II. Métodos inferenciales,* (Ed. Pirámide) and updated versions are available from the address: http://www.aulanet.uniovi.es/ade+/



[Tasas exportaciones] **La Contabilidad Nacional Trimestral elaborada por el INE (http://www.ine.es/tempus/cgi-bin/iti?CODIGO=CNT21A30000) proporciona la información sobre exportaciones recogida en la serieEXPOR.ADE.**

a) **Estudiar la evolución de las exportaciones adoptando como referencia el primer trimestre de 1995.**

b) **Calcular las tasas de variación interanuales e intertrimestrales.**

RESOLUCIÓN

El cálculo de números i

evolución temporal de 1

Trimestres	EXPOR	Ir
1/1995	4030,15	N
2/1995	4160,83	N
3/1995	4233,34	N
4/1995	4307,37	N
1/1996	4424,26	
2/1996	4576,67	
3/1996	4773,81	
4/1996	4985,74	
1/1997	5231,46	
2/1997	5460,96	
3/1997	5656,27	
4/1997	5757,42	

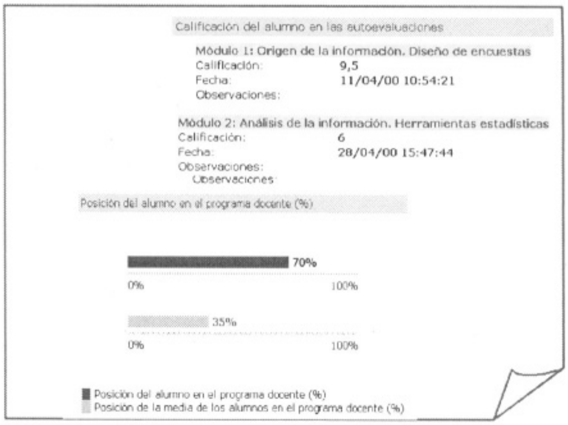
Períodos	Indices	Tasas %
BASE.1/199	1	0
2/1995	1,032426	3,2
3/1995	1,050417	5
4/1995	1,068787	6,9
1/1996	1,09779	9,8
2/1996	1,135608	13,6
3/1996	1,184524	18,5
4/1996	1,23711	23,7
1/1997	1,298081	29,8
2/1997	1,355026	35,5
3/1997	1,403489	40,3
4/1997	1,428587	42,9

Figure 4. The Software ADE+

Virtual teaching success depends to a great extent on interactivity and communication capability. Thus, main attention should be paid to students evolution, including academic, technical and administrative aspects.

In AulaNet, reports are available to students and teachers, providing several useful indicators:

- Student progress in the syllabus, compared with the mean of the group.
- Student participation, obtained as a ratio of the selected interactive options.
- Marks of the practical questions and exercises submitted to tutors.
- Self-Assessment results.
- Results of the final on-line exam.


Calificación del alumno en las autoevaluaciones

Módulo 1: Origen de la información. Diseño de encuestas
Calificación: 9,5
Fecha: 11/04/00 10:54:21
Observaciones:

Módulo 2: Análisis de la información. Herramientas estadísticas
Calificación: 6
Fecha: 28/04/00 15:47:44
Observaciones:
Observaciones:

Posición del alumno en el programa docente (%)

■ Posición del alumno en el programa docente (%)
▨ Posición de la media de los alumnos en el programa docente (%)

Figure 5. Reports about Students

In order to keep a permanent contact with our students, they have repeatedly been interviewed about several aspects of their e-learning experience.

Vaguely speaking, the best results are associated to interactive questions, self-assessment, presentations and another teaching tools. On the other hand, technical difficulties and low speed of the network remain as the most negative aspects.

3. CONCLUDING REMARKS

The AulaNet experience has provided some useful information for the future development of a Virtual Campus at the University of Oviedo.

The implementation of the subject "Economic Data Analysis" has clearly confirmed the demand of e-learning and teaching, suggesting several lines that should be followed in the near future.

Firstly, the development of new virtual subjects and their inclusion in AulaNet should be encouraged, allowing our students to design a more flexible curriculum vitae, with more reasonable timetables and a more participative learning system.

Secondly, new strategies should be designed for the consolidation of a high-quality Virtual Campus in our University. Up to this point AulaNet has been recognised by IBM with the e-business distinction and now we are turning towards the achievement of a quality certification.

Cooperative initiatives with universities and/or institutions should also been encouraged, following the guidelines of Spanish and European academic authorities.

Once we have experienced some of the advantages and risks of these new teaching and learning methods, the challenge of new information and communication technologies must be faced, improving on-line education and developing a high-quality virtual campus.

BIBLIOGRAPHY

1. Aparici R. "Mitos de la education a distancia y de las nuevas tecnologías", Educatión para los medios en un mundo globalizado, UNED, 2000, http://www.uned.es/ntedu/espanol/temas-de-debate/mitos/nuevastecnos.htm.
2, Barone C.A., Luker M.A. "The Role of Advanced Networks in the Education of the Future", Educom Review, 1999, November/December. http://www.educause.edu/ir/library/html/erm9968.html
3. Bauer C., Berkhout J., Chang V., Chin K.L., Glasson B., Tauber J. "Exploring online education: A research framework", in K. Martin, N. Stanley and N. Davison (Eds), Teaching in the Disciplines/ Learning in Context, 28-34. Proceedings of the 8th Annual Teaching Learnig Forum, The University of Western Australia, February 1999. Perth: UWA. http://cleo.murdoch.edu.au/asu/pubs/tlf/tlf99/ac/bauer.html
4. Bricall J.M. Informe Universidad 2000, Conferencia de Rectores de las Universidades Españolas, 2000.
5. Carrasco J.L. "Estado de uso e impacto de la teleeducación en las universidades del mundo", Sociedad Internet 1999-2000. http://www.it.uc3m.es/~jlc/sociedadi/
6. Chalmers J. Virtual Education, 1998, www.musenet.org/~bkort/EdMud.html
7. Facemyer K.C. Virtual Science and Mathematics Fair: Innovative Educational Uses of the Internet and their impact on the Culture of Education, 1996, Washington State University.
8. Lopez A.J., Pérez R., Caso C., Río M.J., Ramos C. "Una experiencia docente en la red. Análisis de datos económicos", I Jornada ASEPELT "Nuevas Tecnologías para la enseñanza de Economía Aplicada, November 1999; Alcalá de Henares.
9. Parker D., Gilding A. "Using the Internet as a teaching tool", in K. Martin, N. Stanley and N. Davison (Eds), Teaching in the Disciplines/ Learning in Context, 302-305. Proceedings of the 8th Annual Teaching Learning Forum, The University of Western Australia, February 1999; Perth: UWA. http://cleo.murdoch.edu.au/asu/pubs/tlf/tlf99/ns/parker.html
10. Pérez R., López A.J. "Los retos de la enseñanza virtual. La experiencia de AulaNet", XIV Reunion ASEPELT-España, June 2000; Oviedo.
11. Pérez R., López A.J., Fernández A.M., Somarriba N. "Tele-enseñanza en la Universidad. El Proyecto AulaNet", Boletín de RedIRIS, 1999; n 50-51: 13-16.
12. Salinas J. "Enseñanza flexible, aprendizaje abierto. Las redes como herramienta para la formatión", EDUTEC, Revista Electrónica de Tecnología Educativa, 1999; 10 www.uib.es/depart/gte/revelec10.html
13. Tiffin J., Rajasingham L. In Search of the Virtual Class: Education in an Information Society, London and New York, 1995.

ED68K
A design framework for the development of digital systems based on MC68000

M. L. López-Vallejo[*], J. M. Fernández Freire[**] and J. Colás[*]
**Dpto. Ingeniería Electrónica, E.T.S.I. Telecomunicación Universidad Politécnica de Madrid Ciudad Universitaria s/n, 28040 Madrid, Spain*

**
I.E.S. de Tomiño, Tomiño, Pontevedra, Spain.

Key words: microprocessor, MC68000, digital systems, programming framework.

Abstract: This article presents a framework for the design and development of microprocessor based systems. The software has been developed to work with a Motorola MC68000 microprocessor based board that is used in the Laboratory of Digital Systems Design of the E.T.S.I. Telecomunicación of the Universidad Politécnica de Madrid, Spain. The framework includes a set of basic tools: a cross assembler, a debugger, an editor, a communication system and on-line help. Experience has proven the usefulness of the framework.

1. INTRODUCTION

Nowadays all information processing systems or telecommunication applications are built on digital architectures with standard processors. Consequently, the graduate students on telecommunication engineering must be prepared to deal with this kind of digital systems. This is the reason why during the first years of the degree several courses and laboratories are conceived to give to the students a deep knowledge on this subject.

In this article we present a framework that has been developed in the Department of Electrical Engineering (Universidad Politécnica de Madrid, Spain) to ease the learning process of the students attending the Laboratory of Digital Systems Design [1]. In particular, we will describe the application called ED68K, which is a framework for the development and debugging of programs running in a Motorola MC68000 board, installed in the Laboratory of Digital Systems. All the students of the Telecommunication engineering degree must attend this laboratory during the third year, being the laboratory the practical complement of a theoretical class where microprocessors are the main subject. Students have to realise a project which consists in the design, implementation and debugging of a digital system. In this laboratory they must deal with both hardware and software problems. The project is made in groups of two, and students work with real equipment driven with the framework we describe in this article. This approach has been followed instead of simulating the design in order to give the students a closer approach to authentic work.

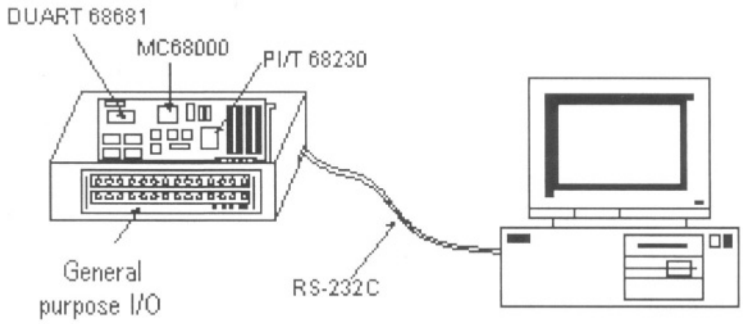


Figure 1 Development environment

The development of this framework started in 1996, when Motorola Inc. gave our department a set of SBC68K boards [2] with educational purposes. Since the boards had no dedicated software, we decided to develop our own interface program. The program features have been growing during the last years, and finally, the program resulted in a complete framework, called ED68K, which includes communication tools, assembler, debugger, etc.

The SBC68K kit is an evaluation and development board for the MC68000 microprocessor manufactured by Arnewsh Inc. The board, which can be accessed by means of a terminal through an RS232-C interface (figure 1), comprises RAM, ROM, timers, a disk controller, a printer interface and general-purpose input/output devices. The ROM includes the *firmware,* which is the software provided with the kit. This software comprises a monitor and a self-contained programming environment with control functions for input/output.

The *firmware* (called TUTOR) is designed to edit and compile the programs in a computer used as host. Thus, the board requires a simple text editor, a cross assembler and a program that emulates a terminal to load the object code in the board and to debug it. Obviously, if we compare this working environment with the power and robustness of the current programming frameworks it is clear that this scene can be drastically improved. A first idea we had was just to provide a user interface making good use of all the graphical resources provided by *Windows*. Later on, the resolution of different problems that appeared in the development process resulted in a complete framework that was built to help students (and designers) in the whole design cycle of digital systems. The product of this work is ED68K, the application that will be described in the following sections.

The structure of this paper is as follows. First, the main goals of the framework will be described, as well as the implementation methodology that has been followed. Next, the framework is described in depth: the graphical interface, the assembler, the debugger and the communication mechanisms. Finally, other features of the software will be presented and some conclusions will be drawn.

2. FRAMEWORK GOALS

The main goal of the development of ED68K was to provide an interface to the SBC68K board that could simplify the designer work. In this way, the program was defined as a framework that integrates a cross-assembler, a code editor, a debugging environment and a communication system. Main low level tasks integrated in the framework are:

1. Compilation of assembler programs.
2. Load of object code in the RAM memory of the board.
3. Execution of programs from a given memory address.
4. Setting up breakpoints.
5. Execution of programs step by step.
6. Display and modification of the content of the registers.
7. Display and modification of the content of the memory.

In addition, since the program was designed to work in an undergraduate laboratory, a secondary objective was considered: to assist the students in their learning process. In this way, the framework has a comprehensive on-line help, display facilities (to see the contents of the microprocessor registers, for instance) and a printed user's manual with a tutorial [3].

3. IMPLEMENTATION APPROACH

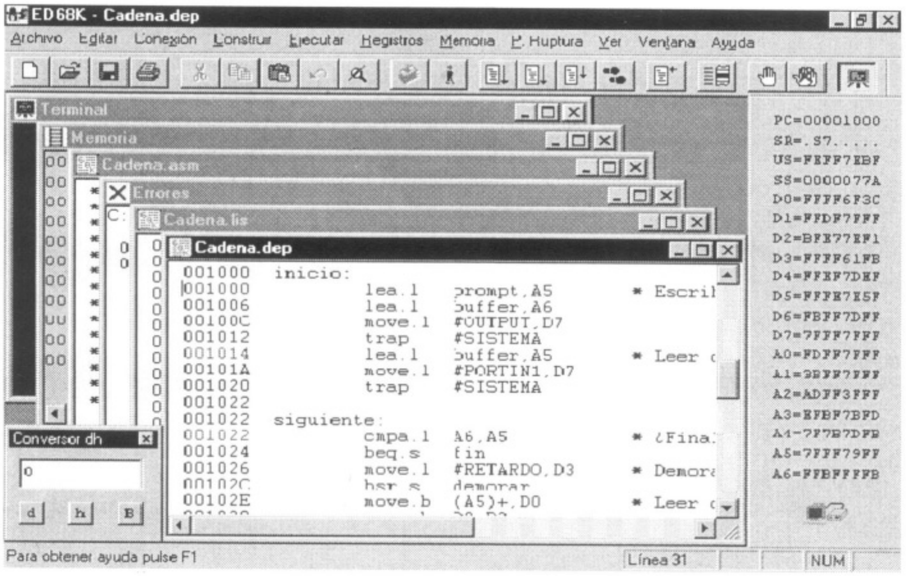
ED68K has been developed under *Windows*, because this was the operating system available in the laboratory machines. Besides, students are very familiar with this operating system, (even though there are more powerful operating systems, like *Linux*, but still with very few users).

As programming language we have chosen C++ [4] to follow the *object oriented programming methodology*. Moreover, there are many C++ libraries to develop *Windows* applications whose use eases considerably the development process. An important issue we had in mind when developing ED68K was to generate *reusable code*. Taking this programming style into account, the final implementation can be easily adapted to another hypothetical evaluation kit that would follow the same connection style and features as SBC68K. Additionally, we can take advantage of two important characteristics of the object oriented programming: encapsulation and inheritance.

Nowadays, there are different options to develop *Windows* applications using C++ as programming language. For instance, the Object Windows Library (OWL), from Borland, or the Microsoft Foundation Class (MFC). Both provide high level classes that allow the programmer to build an initial skeleton. In particular, the application that is described here has been elaborated using as support the Microsoft Foundation Class (MFC [5] version 4.2), following the *document-view* architecture.

ED68K is a document oriented application that has been designed following the MDI model *(Multiple Document Interface)* [6]. The documents the application handles are the different files the framework employs: source files, object files, debugging listings, etc. This model allows the user to simultaneously edit several source files, which improves the framework flexibility and eases the debugging process.

The application has been developed making good use of another interesting feature: the usage of *threads* to improve the framework performance. In the Win32 API there are no different threads: the API only needs to know the starting point of the threads to begin with their execution. However, MFC has two kinds of threads: user interface threads and working threads. On the one hand, interface threads are usually employed to process the user input, answering the events and messages generated by the user. This kind of threads is able to create new windows or process the messages the windows receive. On the other hand, working threads are designed to work in background not requiring user interaction.


ED68K - Cadena.dep
Archivo Editar Conexión Construir Ejecutar Registros Memoria Ruptura Ver Ventana Ayuda
Terminal
Memoria
Cadena.asm
Errores
Cadena.lis
Cadena.dep
Conversor dh
d h B
Para obtener ayuda pulse F1
Línea 31
NUM

```
001000   inicio:
001000            lea.l    prompt,A5       * Escri
001006            lea.l    buffer,A6
00100C            move.l   #OUTPUT,D7
001012            trap     #SISTEMA
001014            lea.l    buffer,A5       * Leer c
00101A            move.l   #PORTIN1,D7
001020            trap     #SISTEMA
001022
001022   siguiente:
001022            cmpa.l   A6,A5           * ¿Fina
001024            beq.s    fin
001026            move.l   #RETARDO,D3     * Demora
00102C            bsr.s    demorar
00102E            move.b   (A5)+,D0        * Leer c
```

```
PC=00001000
SR=.S7.....
US=FEFF7EBF
SS=0000077A
D0=FFFF6F3C
D1=FFDF7FFF
D2=BFE77EF1
D3=FFFF61FB
D4=FFEF7DEF
D5=FFFE7E5F
D6=FBFF7DFF
D7=7FFF7FFF
A0=FDFF7FFF
A1=3BFF7FFF
A2=ADFF3FFF
A3=EFBF7BFD
A4-7F7B7DFB
A5=7FFF79FF
A6=FFBFFFFB
```

Figure 2. ED68K Graphical Interface

ED68K is a multi-thread application with two threads: a user interface thread that supports the user interaction and a working thread that manages the communication with the kit during the execution of the user programs.

4. DESCRIPTION OF THE FRAMEWORK

The framework is characterised by the complete integration of all the tools. Visual integration has been accomplished since there is a common user interface. The framework has also data integration, because editor, assembler and debugger share all the files. Moreover, there is control integration, given that some tools are activated by some others.

4.1 Graphical Interface

A snapshot of the user interface of ED68K is shown in figure 2. In this figure we can see most of the elements the graphical interface is composed of. On top of the main window is placed the Title Bar, with the name of the application and the name of the file that is currently opened in the text editor (as all the figures, in Spanish, Cadena.dep).

The Menu Bar, the area displayed across the top of a window directly below the title bar, displays the names of the menus. Clicking a menu name displays a list of commands for accessing ED68K functions. Available menus are (from left to right in figure 2): File, Edit, Connect, Build, Run, Registers, Memory, Breakpoints, View, Window and Help. Below the menu bar is the Toolbar, which provides quick, one-step access to 20 commonly used features.

In the central area we can see six cascaded windows. Three of them correspond to three files opened within the editor: a source file (Cadena.asm), a listing file (Cadena.lis) and a debugging one (Cadena.dep). The `Terminal` window is used to display the input/output of the user's programs. The `Error` window displays the errors produced while assembling, and the `Memory` window shows a memory dump.

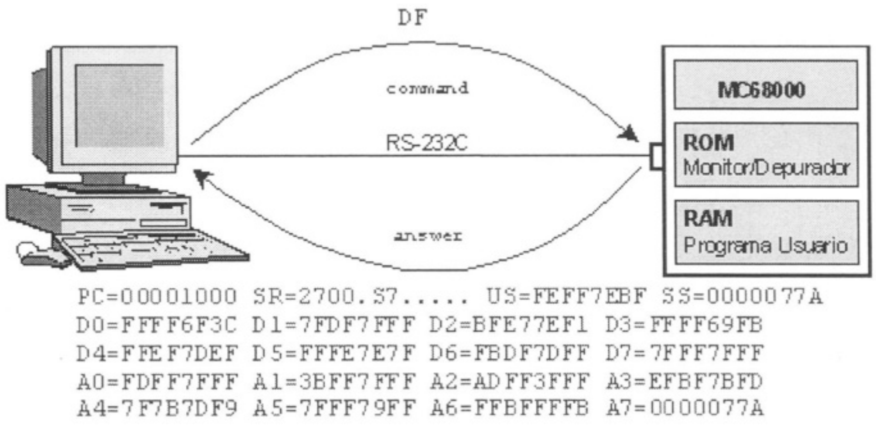
On the right of the previous windows the Register Bar appears with the purpose of displaying the contents of the registers of the CPU on the board and to interface their modification. On the left hand side a useful hexadecimal to decimal (and vice-versa) converter is shown. Finally, at the bottom of the screen is the Status bar, which gives information on the selected object or on your current action.

4.2 Assembler

In order to better control the assembling process we decided to integrate a cross assembler within the framework, instead of using the one provided by the *firmware*. We used as starting assembler the one that was done by Paul McKee, whose source code was available on the Internet [7]. Having the source code, we were able to modify many different points:

– To replace the user interface (originally it was a command line).
– To integrate the assembling process within the development process.
– To ease the source code debugging.
– To translate the assembler error messages.

After all these actions had taken place, a new version of the assembler was built as a DLL (Asm68k.dll). This DLL format was established to maintain the original structure and to ease the replacement of the assembler if necessary (again, the framework has been designed reusable).



```
PC=00001000 SR=2700.S7..... US=FEFF7EBF SS=0000077A
D0=FFFF6F3C D1=7FDF7FFF D2=BFE77EF1 D3=FFFF69FB
D4=FFEF7DEF D5=FFFE7E7F D6=FBDF7DFF D7=7FFF7FFF
A0=FDFF7FFF A1=3BFF7FFF A2=ADFF3FFF A3=EFBF7BFD
A4=7F7B7DF9 A5=7FFF79FF A6=FFBFFFFB A7=0000077A
```

Figure 3. Running a program: updating the register bar

Significant modifications we can mention here were the elimination of all the input parsing, the generation of object, listing and debugging files, or the cooperation between the assembling process and the load of object code into the board. The assembling errors are displayed on the Error window and on the listing file. The source lines that caused these errors are listed after them. The object file is generated with S-record format, which is compatible with the kit SBC68K. We have also redefined the assembler directive END in such a way that the debugger can automatically set up the program counter with the first address of the compiled program.

4.3 Running and debugging programs

The execution of user programs takes place in the SBC68K board. The framework uses the firmware debugging primitives to create the commands offered by the user interface of ED68K. The board produces as response to every single primitive a string of characters parsed by the framework and is finally displayed on the user interface objects (message boxes, register bar, terminal window, etc.). In this way, the command-based interface of the kit is converted into a Windows-based graphical interface. This procedure is described in figure 3, where as example an update of the register bar takes place as an example.

The final debugger of ED68K integrates four execution modes:

– Without breakpoints.
– With breakpoints.
– Until the current instruction (the line marked with the caret in the editor).

– Step by step.

Whenever the user is debugging a program, the Register bar is updated after every execution command. Besides, in the debugging window the instruction pointed out by the program counter is emphasised.

The `Terminal` window displays the input/output of the user's program. Several questions must be considered to correctly perform the I/O operation:

– The execution of programs can take a long time, thus delaying the board response. Nevertheless, the user interface must be able to provide a quickly response not relying on the board status. Therefore, while the kit is running a user program, the framework cannot be stopped[1].

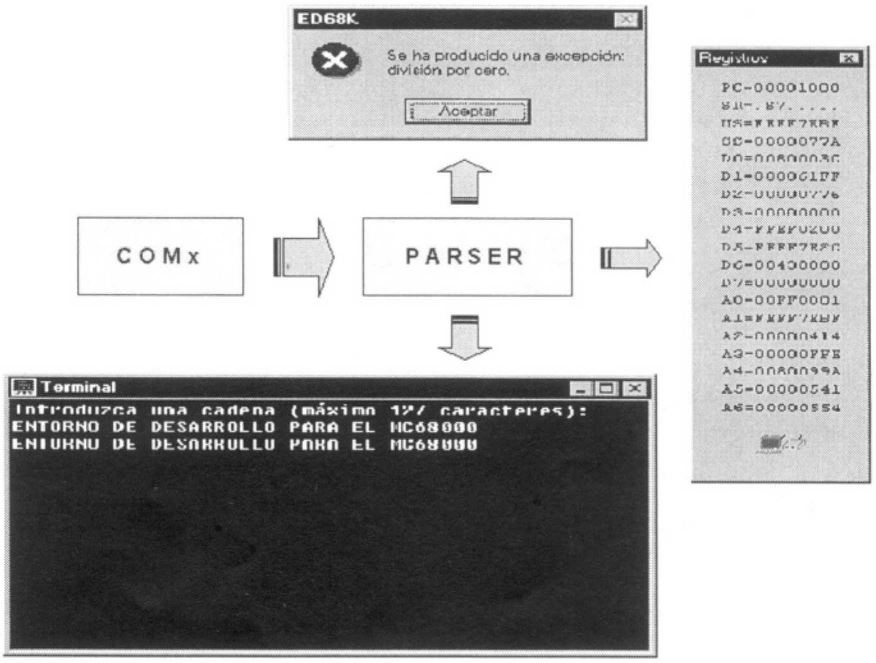
– To implement the running modes "*With breakpoints*" and "*Until the current instruction*" the framework analyses the trace returned by the board. From this trace we can extract the value of the registers (to be displayed on the register window) and the address of the next instruction to be executed (to be supplied to the debugging window).

– While the programs are running, some error situations might happen. In this case the MC68000 produces an exception and the kit sends an error message to the terminal followed by an information trace. The best way of processing this situation is by means of an error message box. Consequently, errors are not displayed on the Terminal window in order to keep the output of the user program clear.

The previous considerations have induced two design decisions on the framework debugger:

1. We have used a working thread to read the user program output from serial port and display it on the Terminal window. This thread is created when the execution of the program is started, and is destroyed after completion of the program.
2. The information produced by the kit is parsed to detect the messages sent by the board (errors, stops at breakpoints, etc.) while running the programs. These messages will be processed by message boxes, avoiding their presentation on the Terminal window (as it is shown in figure 4). Following this procedure, the framework assists the designer in an efficient way.

[1] The situation would be even more critical if the user program is hanged



```
Se ha producido una excepción:
división por cero.
```

```
Introduzca una cadena (máximo 127 caracteres):
ENTORNO DE DESARROLLO PARA EL MC68000
ENTORNO DE DESARROLLO PARA EL MC68000
```

```
PC-00001000
SR-.S7.....
US=FFFF7FFF
SS-0000077A
D0=0080003C
D1-000061FF
D2-00000776
D3-00000000
D4-FFEF0200
D5-FFFF7FFC
D6-00400000
D7=00000000
A0-00FF0001
A1=FFFF7FFF
A2-00000414
A3-00000FFE
A4-0080099A
A5-00000541
A6=00000554
```

Figure 4. Process followed to display the user's I/O

4.4 Communication with the board

Due to the fact that the programs run on the SBC68K kit, the ED68K framework establishes a communication protocol with the board firmware to load the programs on the RAM, and to run and debug them. The communication takes place through the serial port, and includes:

– sending commands to the board monitor and reception of response.
– loading programs on the kit memory.

Since the framework was conceived to have the assembler and editor working independently, the application does not require having the kit connected to the PC from the beginning. Thus, the users can edit and compile programs even if there is no SBC68K kit available. On the other hand, the kit is absolutely necessary to debug and run programs. In this case the user must configure and open the serial port before starting the working session. Additionally, the debugger also uses two elements of the user interface: the Register bar and the Terminal window. Consequently, those objects are only created once the user establishes communication with the board.

A special flow control protocol has been implemented to properly communicate the framework with the board. This protocol was necessary to avoid overflow problems when transferring the object programs to the kit. Excellent performance has been obtained with this protocol, improving classical terminal emulation programs in 20%.

4.5 Other features

ED68K includes other features, as a text editor, an on-line help, a simple hexadecimal-decimal calculator or an installation program. Space requirements preclude us from commenting further on these and many other interesting features of the framework. Thus, only some details will be presented next.

A text editor has been designed and integrated in the framework allowing the edition and printing of files. It was specially developed to facilitate the cooperation among the assembling and debugging tools. In this way, the editor has been designed in such a way that the user can only modify source files (.asm), leaving untouched listing (.lis), object (.h68) and debugging (.dep) files, which can just be read. Another interesting feature of the editor we can mention here is the automatic indentation of the text, which helps the designer both coding and debugging source files. Additionally, standard editing functions are supported, such as: handling of source, listing, object and debugging files; select, copy and paste text fragments, search and replace, etc.

On-line help was created with the application Help Workshop supplied by Visual C++. ED68K provides help at three levels: Help Contents, Context Sensitive Help and ToolTips. Help topics are shown in a dialogue box with three folders: Contents, Index and Find. These are three different ways of accessing the help information. The Context Sensitive Help (or ScreenTips) shows information about the different elements on the screen. ToolTips are small information boxes that are displayed when the mouse pointer stops on a particular element of the user interface.

The framework also has a simple calculator, which converts hexadecimal into decimal representations and vice-versa. This tool is especially useful when working at assembler level. The converter can handle 32-bit numbers, and is completely integrated into the application: it allows transferring data from its display to the different edition boxes via the clipboard.

Finally, the installation program was developed using the public domain tool Inno Setup 1.1 [8] because of the excellent performance it provides. This tool can be executed under Windows, being the source code available. Interesting properties of Inno are that the installation program can be put in a single file and de-installation can be fully accomplished.

5. CONCLUSIONS

This article has described a framework developed for the design of microprocessor-based systems with educational purposes. This framework provides the ideal environment to work with MC68000 microprocessor, including editing, assembling, debugging and execution tools.

The framework has been widely used for the last two years in the undergraduate course *Digital Systems Laboratory* (Degree on Telecommunication Engineering, Universidad Politécnica de Madrid). The tools included in ED68K have shown excellent results as learning environment and microprocessor-based design assistant.

An additional advantage of the framework is that the resulting code is completely reusable. In this sense, future work covers adapting this framework to a different evaluation kit: a board based on the MC68331 micro-controller.

ACKNOWLEDGEMENTS

The authors would like to thank J. J. Montero for his valuable help preparing the presentation of this tool at the SIIE'2000 Conference.

REFERENCES

1. Studies Program of the Telecommunication Engineering Degree: Resolución de la Universidad Politécnica de Madrid. BOE 31 May 1994. http://www.etsit.upm.es/docencia/grado/planl994/planl994.html
2. Arnewsh Inc. SBC68K User's Manual. 2nd Edition. 1991.
3. J. M. Fernández Freire and M. L. López Vallejo. Entorno de desarrollo ED68K. Manual de usuario (V 1.1). Publicaciones E.T.S.I. Telecomunicación, Feb. 1999.
4. B. Stroustrup. The C++ Programming Language. Third edition. Addison-Wesley, 1997.
5. J. Prosise. Programming Windows 95 with MFC. Microsoft Press, 1996.
6. Microsoft Corporation. The Windows Interface Guidelines for Software Design. Microsoft Press. 1995.
7. Paul McKee, www.ticalc.org/pub/dos/asm.
8. Jordan Rusell Inno Setup 1.1 www.jordanr.ml.org

HCI Curricula in Spain

A cooperatively designed, free Web-access syllabus

J. Lorés*, J. Abascal, J.J. Cañas, I. Aedo, M. Gea, M. Ortega, L.A. Ureña, P. M. Valero, M. Velez

**Computing Department, Universitat de Lleida, C/ Jaume II 69, Lleida*
Tlf: 973.702.700 Fax: 973.702.702
e-mail:jesus@eup. udl. es

Key words: virtual teaching, human-computer interaction, CSCW

Abstract: In this paper we present an initiative born within the Spanish Human-Computer Interaction Association (AIPO). Our aim is the creation of a groupwork to discuss the basic contents of an introductory course to HCI and to design a common-core syllabus of curricula contents in Human-Computer Interaction (HCI) in Spanish, using the Internet as the medium of sharing contents. It is an initiative to promote and allow a free access of HCI curricula to the Spanish community and the Latin American community. It is open to all interested contributions from any educational institution. As an example we present the experience of the application of the virtual course *Introduction to HCI* made at the Universities of Lleida, Granada, Carlos III and Castilla-La Mancha.

1. INTRODUCTION

The creation of the Association of Human-Computer Interaction (AIPO) in Spain in November 1999 meant a strong impulse to the emergence of HCI within the Spanish universities. The objective of our association is to establish links between the teachers interested in this area and to promote research and the creation and improvement of teaching materials. Education, in our case, is a subject to which we have to dedicate an important effort, due to the lack of this type of material in Spanish. AIPO just was the place to launch debates and contacts on the educational aspects of HCI. Its initial meetings gave birth to the idea of working together in the creation of common educational contents instead of having each university working separately. A logical proposal from these talk gatherings was the use of the Internet as the support for this shared information. In addition, it was decided to offer it for free in order to facilitate the open access to all the Spanish-speaking community. Within this framework, the initiative was fixed in the preparation of a course of Introduction to the Human-Computer Interaction.

2. VIRTUAL EDUCATION

Virtual tools make it possible to combine the learning process based on peer-to-peer classroom activities with continuous training at any moment, enabling the reception of information, materials, resources, experiences.

In order to develop virtual training in an effective way, technical capacities related to Web and to interactive tools as well as pedagogical and organisation capabilities will be required.

3. A JUSTIFICATION OF TEACHING OF HUMAN-COMPUTER INTERACTION

The Report of ACM/IEEE-CS Joint Curriculum Task Force Computing Curricula of 1991 [1] establishes nine thematic areas to cover the computer science discipline. Human-Computer Interaction is one of these areas in ACM curricula, an important aspect to set up this discipline in the curricula of the Spanish Universities. In 1988, the SIG in Computer Human Interaction (ACM-SIGCHI) started a group with the aim of making a curricula framework [2]. The task of the committee was to write a series of recommendations for education in HCI. This committee wrote the document ACM SIGCHI for Curricula in Human-Computer Interaction in 1992 presenting the recommendations for the accomplishment of HCI courses.

In order to be able to cover all the aspects, their definition and the objectives, HCI has to take into account different areas concerning the human being and the computer: Computer science (design and engineering of the interfaces), psychology (theory and application of the cognitive processes and the empirical analysis of the users' behaviour), sociology and anthropology (interaction between technology, work and social organisations) and industrial design (interactive products).

The subjects chosen in the ACM curriculum evolved from the consideration of the interrelated aspects of Human-Computer Interaction:

(N) Nature of the interaction, (U) Use and context of computers, (H) characteristics of the Human being, (C) Computers and architecture of the interface and (D) Development process. It is also necessary to consider the (P) Presentation of projects and the evaluation.

4. COURSE CONTENTS

Currently the course consists of thirteen chapters. The first one is an introduction to Human-Computer Interaction, where it introduces the discipline, the concept of interface and its usability.

The second and third chapters try to present the two participants in the interaction: the computer, with the study of the devices, and the human being, as an interactive agent. The interaction styles are a fundamental chapter focused on a comparative perspective of the interaction paradigms and studies. Next, the study of metaphors and their use in design is presented.

The presentation of a study on user support and styleguides helps to complete the training in interaction models.

Internationalisation is not a common chapter in the bibliography, but we consider it fundamental in the current globalised context of interaction.

The next section focuses on design, with a chapter dedicated to the interface engineering, other dedicated to development tools, graphical design and to the evaluation of the interfaces. A final section is devoted to groupwork interfaces.

5. THE COMPLETE PROJECT

Our vision is to combine face-to-face and virtual work, reproducing, in the virtual world, most of the facilities we have in the real world, adding up interactivity, the most important feature in virtual teaching.

The final virtual support to the teacher will consist of the course, the slides, a video recording/library with a class for each chapter done by different experts, some interactive exercises written in Java and a self-evaluation test.

In the following paragraphs we will present the course contents and the slides developed up to now as a website to present these contents in the classroom.

5.1 The virtual course

The present structure of the course is a Website located in the server of the Griho group of the University of Lleida (http://griho.udl.es/ipo) that contains all the documentation of the course and the slides corresponding to each chapter. The design of the course follows a structure of three parts as it is seen in Figure 1.

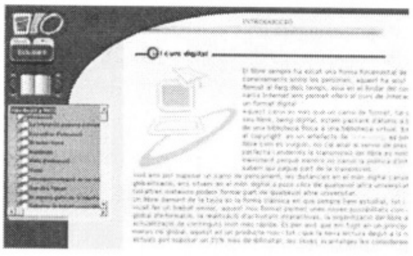


Figure 1. The Virtual Course

Once the student enters the digital book, a webpage appears divided in three parts: frame number 1 on the left corresponds to the navigational area, frame number 2 in the upper part covers the synthesis that means the context information, and frame number 3 is the contents area.

Each chapter has the following parts: an introduction, objectives, different levels of contents, bibliography and selected links and exercises.

5.2 The slides

The slides corresponding to each chapter of the course are developed in Microsoft PowerPoint. With the help of a projector, it will guide the presentation of the different topics of the course to the students.

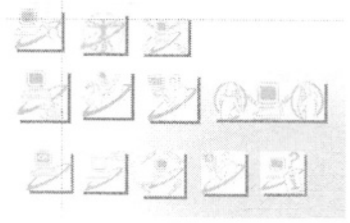


Figure 2. The Slides

For the accomplishment of the course, the Griho group have been working in its preparation. They are a group of teachers of the AIPO association with experience in diverse parts of the agenda: Psychology, design, formal groupwork, methods, etc. The course allows the inclusion of documentation through the modification of the index, a text type file, visualised by means of a Java applet. The contents have been discussed using electronic mail and they have been converted to HTML format, by the facility that involves the use of groups. For the writing of the contents we have used bibliographical resources by Preece [3], Dix [4] and Shneiderman [5]. They have been used for the accomplishment of the course, as well as abundant material from the Internet and from the ACM digital library.

6. APPLICATION AT THE UNIVERSITY OF LLEIDA

The course has been applied in the current teaching of the subject Introduction to Human-Computer Interaction at the University of Lleida, (located in Lleida, Catalonia about 160 km far from Barcelona) in the subject "Human-Computer Interaction Systems" included in the "Interactive Technologies" syllabus for the Computer Science studies.

The subject is scheduled in two theory hours and two applied, practical lab hours a week. For the theory classes we used a general classroom for all the students, but groups with fewer students would be desirable. In these classes we used the laptop connected to the Internet, the tube projector and a video player. As presentation material, they use transparencies based on the digital course. The scheme followed is a presentation of the subject, the objectives to obtain, an explanation of the subject contents and finally the conclusions.

As a complement, video shows relative to the topics are displayed. Finally a list of the bibliography resources and the links to related Websites are given. As supporting educational material is available on the Internet, the students can consult the digital course online or download it to their computer to have it off-line. A paper copy is another facility offered to the students, to reduce the total time watching the screen.

7. APPLICATION AT THE UNIVERSITY OF GRANADA

The digital book has been used at the University of Granada to teach courses in three curricula: Graphical Design at the School of Fine Arts, Ergonomics at the Psychology Department and User Interface Design at the Computer Science Department.

Lectures and students have shown significant improvement in their performance by using the book and the opportunities it offers to them.

The book has been used as supportive material to teach HCI concepts and tools to students with different backgrounds. The students participated actively in regular seminars in which several issues were discussed. They had the opportunity to interact with the authors, inside and outside the University of Granada, through the Internet. In that way, they took advantage of having lecturers with a different background, which complements the understanding of these concepts. Students from different courses could also interact among each other sharing experiences and knowledge.

Having the book available on the Internet allowed overcoming the limits of the temporal and spatial dimensions of traditional University courses. Students could communicate among each other and with the lecturers at any time of the day and from any place (a cyber-café, home, via WAP, etc.). In this way, several researchers referred in the book have been invited to the University to cover relevant issues of the book (for example, this year Gerrit van der Veer was invited to give a conference about group task analysis). It should be mentioned that both students and lecturers have expressed a great enthusiasm and satisfaction with the book and their comments will be very helpful to improve new versions of the book.

8. APPLICATION AT THE UNIVERSITY OF CARLOS III

Since 1995, we have been teaching a course in the CS Engineering degree at the Carlos III University called "Hypermedia Systems: Design and Evaluation". It is aimed at covering the theoretical and practical lacks that our students have concerning hypermedia systems and applications. In this course, which has been taught since then in the sixth and last semester of this 3-year degree, we introduce our students to the application of software engineering methods to produce usable and useful hypermedia applications. In particular, we focus the classes in the development of a hypermedia prototype following a specific method named Ariadne [6]. It is centred on designing, implementing and evaluating hypermedia systems applying basic principles of software engineering and HCI. In particular, we consider that teaching HCI principles, which are essential to develop any kind of interactive systems, makes it possible to achieve an integral education process, since as stated by Friedman and Kahn [7], HCI studies build a bridge between social and technical issues.

For this reason, basic principles to design usable user interfaces and tools to provide an effective navigation in large information spaces are studied. Students are reminded of the fact that they are not expected to test their ability as programmers but their ability as software developers, who are capable of creating useful software applications whose intended users may have no technical background. Moreover, evaluation methods and techniques are proposed as the basis for an iterative design aimed at involving users in the development process. This activity is basically oriented towards communicating them the idea that in their work as software developers they will have to take into account the users' needs and knowledge about the tasks domain. In the information society, almost everybody will rely to some extent on computers to perform daily tasks, such as their work, entertainment, leisure, commerce or education. Therefore, software applications supporting such processes have to be usable from the point of view of their users.

Our impression after these teaching experiences is quite encouraging, since most of our students use the hypermedia and HCI methods we teach not only in the tasks we ask for but in other projects developed within a different research group.

9. APPLICATION AT THE UNIVERSITY OF CASTILLA LA MANCHA

This digital book has also been used at the Escuela Superior de Informática, Universidad de Castilla La Mancha in the subject called “Computer-Human Interaction Systems” in the year 2000-2001. This course is studied at the 4th year in the Computer Science Engineering degree. Providing 9 credits, the course is given in the first term and consists of 6 hours a week, 4 hours for theory and 2 hours for lab practicals. As it is quite a long course, additional chapters have been added to the book, mainly about topics related to direct manipulation, information search and representation.

Since there is a specific profile on HCI in the Higher Grade (4th and 5th years), some topics on learning and collaborative environments have also been added. In this Profile there are other six subjects about Interaction, which are optional and set a study line leading to this specific domain.

The course has been developed in a similar way to the one referred above for the University of Lleida. In the same way the students explicitly prefer the printed course to the digital book. For this reason we consider very convenient having the book printed.

10. CONCLUSIONS

The experience obtained in the preparation of the course has been very positive because of the benefits derived from the creation of educational contents by a group of teachers of different universities, experts in different aspects of the topics of the course. The use of the Internet, therefore, has been fundamental to its accomplishment through the use of the electronic mail, mailing lists and file transfers. We are taking into consideration the development of a workgroup tool that will allow uploading and simultaneous updating of new or revised contents.

The educational experience of the use of the digital course at the University of Lleida has been very positive for the last two years. It was mainly so positive last year with the design of the digital course in group. However, it is interesting to comment the still on-going need to print the book in paper because of the difficulty imposed by the standard technology to directly present all the documentation on the screen.

REFERENCES

1. ACM/IEEE-CS (1991) joint Curricullum Task Force Computing Curriculla 1991.
2. Hewett, T. T (1992). ACM SIGCHI Curriculla for Human-Computer Interaction.ACM
3. Preece, Jenny (1994). Human Computer Interaction. Addison and Wesley.
4. Dix, A, Finlay J., Abowd G., Beale R. (1998). Human-Computer Interaction 2ª ed. Prentice Hall, 1998. LSI n° 1119. ISBN: 0-13-239864-8
5. Shneiderman, Ben (1998): "Designing the user Interface: strategies for effective human-computer interaction, 3ª ed.". Addison Wesley, .LSI n° 29. H5 SHN des.
6. Díaz, P., Aedo, I., Panetsos, F. (1999): A methodological framework for the conceptual design of hypermedia systems. Proceeding of H2PTM. París, September 1999. pp. 213-228.
7. Friedman, B. and Kahn, P.H. Educating computer scientists: Linking the social and the technical. Comm. of the ACM 37,1 (January 1994), 65-70.

Interactive Design of Adaptive Courses

J. A. Macías and P. Castells
E.T.S. de Informática, Universidad Autónoma de Madrid

Key words: authoring tools, interactive design, software, task modelling, user modelling, distance learning, human-computer interaction

Abstract: The ability of educational applications to adapt to individual students has been a sought-after feature for more than a decade. The difficulty of developing this kind of software and the lack of adequate tools that facilitate this endeavour have hindered the participation of instructors and teachers in the elaboration of adaptive applications. In this paper we describe an authoring tool, ATLAS, that aims at combining expressive power and ease of use in the design of adaptive web courses. ATLAS allows the fully interactive construction of courses that adapt automatically to the student's characteristics and her/his behaviour while taking the course. The designer interacts with the tool by using an intuitive visual language based on the direct manipulation of elements involved in the course. The tool takes care of the transition between a teacher's understanding of the course and the representation model of the underlying system.

1. INTRODUCTION

With the rapid introduction of the WWW technology in the educational field, educational applications are facing an increasingly wider and more heterogeneous audience. In this context, the diversity of needs and particularities in such large communities of learners is becoming a first-order concern. Since it is impossible to satisfy the preferences and needs of all students with a single solution, attempts are being made to produce flexible software, able to adapt to different kinds of users, platforms, and situation. Given its complexity, this kind of software is extremely difficult to develop, which hinders the direct, active involvement of education professionals in the elaboration of the product, to the detriment of its pedagogical quality.

A wide variety of interactive authoring tools and easy-to-use graphical environments are available today that support diverse aspects of courseware authoring, like WebCT [1], Macromedia Director, or HTML editors, to name a few. These tools greatly simplify the development of educational materials, but they do not provide an adequate support for the dynamic aspects of an adaptive course. Moreover, the type of assistance they provide tends to be confined to work over partial aspects of a course, or a subset of aspects, in isolation, like multimedia components, simulations, static tables of contents, or web-site administration, in such a way that the dynamic integration of all these elements is done by using supplementary non-interactive tools or programming languages.

On the other hand more powerful tools of a different kind have been developed during the last few years that do support the construction of interactive courses with a high degree of flexibility, including adaptive capabilities, using a relatively small set of high-level primitives [2, 3, 4]. However these tools are not easy to use in general, as they require learning special-purpose specification languages, and understanding non-trivial abstract concepts that are out of reach for authors who are not necessarily familiar with the complexities of software development. Complementing this type of tools with editors based on intuitive visual languages, to isolate the designer from the complexity of the underlying system, is a difficult problem when building dynamic courses, since the course as such does not exist at design time.

In this paper we describe a design tool, ATLAS (Authoring TooL for Adaptive Software design), for the interactive construction, in a graphical editing environment, of interactive courses with the capability to dynamically adapt to students. ATLAS helps courseware authors get a better understanding of the different aspects of an adaptive course and the relations among them: structure, contents, presentation, and student model, by integrating the design of all these elements in a single visual environment. To reduce the need for abstraction from the author, the tool offers the option of working at design time on a representation of courses that is close to the final result, by using concrete examples of the student model, simulated by the designer.

Our system aims at bringing the advanced techniques for adaptive course construction recently developed in the field of computer-assisted learning within reach of instructors and subject-matter experts that do not have a strong computer science background. ATLAS combines the expressive power of a system based on the abstract modelling of the student and his/her learning tasks, with the ease of use of graphical specification based on direct manipulation, relieving the designer from using textual specification languages.

ATLAS is being developed on top of a pre-existing tool, TANGOW [4], that is used as a support system for modelling and executing the adaptive courses defined by the designer in the ATLAS environment (see figure 1). The current version uses the WWW as the platform for the execution of the courses.

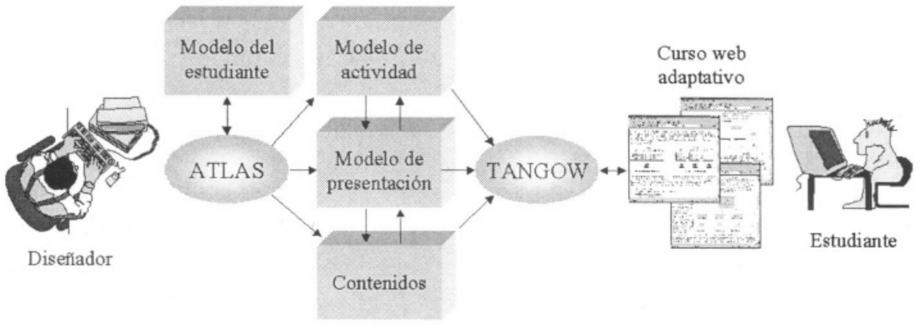


Figure 1. System architecture

2. RELATED WORK

Adaptive hypermedia support technology has made a considerable progress during the last decade, in particular in the field of Computer-Assisted Learning, with systems such as DCG [2], ELM-ART [3] and TANGOW [4]. ELM-ART and DCG use an explicit representation of domain concepts, interrelated in a prerequisite graph. DCG includes a planner that guides the student along a path to reach goal concepts starting from already known concepts. ELM-ART uses a sophisticated system to estimate the knowledge acquired by the user in relation to a concept map of the course, according to which the system dynamically proposes the student a path to follow at each moment. While in ELM-ART and DGC the structure of courses is fixed, being the student itinerary that varies, TANGOW generates the course structure at runtime. TANGOW models the student activity in the form of a hierarchy of tasks that represent didactic units the student can perform. Contents are associated to tasks by assigning a set of HTML fragments to each task, from which web pages are generated when the student is ready to undertake a certain task. Hierarchy among tasks establishes that the accomplishment of certain tasks consists of carrying out other subtasks, which in turn may be composite or not. The structure of the task tree is dynamic and depends on student characteristics and her/his behaviour while interacting with the course.

From the point of view of the designer, adaptive course modelling in TANGOW can be compared to the description of user tasks in model-based user interface construction tools like UIDE [5], MASTERMIND [6] and MOBI-D [7]. Aimed at a wider scope beyond the educational domain, these systems support the construction of user interfaces from a modelisation of the tasks that the user must be able to carry out with the application being built. Both UIDE and MOBI-D include facilities for the graphical manipulation of user tasks. MASTERMIND does not provide a visual editing environment for task models, but provides a very powerful specification language for tasks. The main novelty in our work with respect to these systems is the explicit manipulation of a user model in the development environment itself.

Relating user models to task models requires adequate abstraction capabilities from the specification language. Dealing with abstraction in a graphical environment raises new difficulties. Visual languages are appropriate to transmit declarative information, but it is not easy to describe dynamic behaviour this way. Programming by Example systems [8] solve this problem by inferring procedural information from concrete examples created by the designer. We have carried this idea to our context, in particular the techniques used in HANDSON [9], a tool that allows the visual construction of dynamic interface presentations that depend on any parameter of the application, the platform of the interface itself.

3. THE ATLAS AUTHORING TOOL

ATLAS takes care of the transition between the teacher's view of the course and the representation model of the underlying system. In this sense ATLAS can be considered as a user interface for the course designer in TANGOW. The development environment presents all the elements of the course to the designer (tasks, contents, student model) as objects on the screen that the designer can create, manipulate, and associate to each other directly with the mouse. Figure 2 shows an intermediate stage in the creation, in this environment, of a course about Java programming. The workspace consists of three main areas: task model, contents model and student model. In the first one (left side), the designer defines the structure of the course by building a model of student tasks which, in accordance with the TANGOW model, are organised hierarchically. The contents area (upper right) shows a catalogue of HTML fragments that store the contents of the different parts of the course. In the student model area (lower right), the student characteristics that are relevant for the course being constructed are described. This model consists of a set of simple and composite attributes, forming a tree, that the designer can edit and use for the definition of conditions that will determine the course structure.

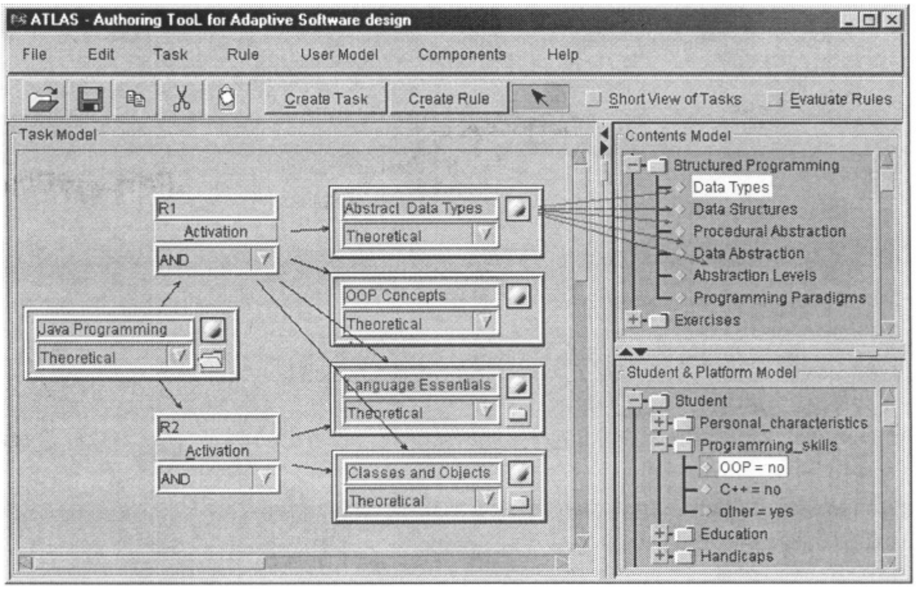
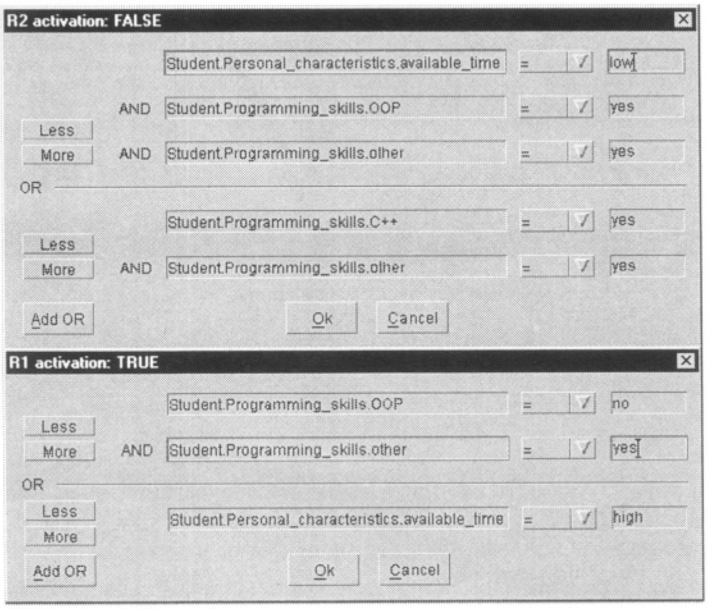


Figure 2. The ATLAS design environment

The designer assigns each task the content units that correspond to it, by dragging objects with the mouse from the course contents area onto the corresponding task. The task tree is constructed by dragging the mouse between tasks, associating each task with its subtasks. In figure 2, the designer has decomposed a high-level unit (task) about *Java Programming* into four subtasks: *Abstract Data Types*, *OOP Concepts*, *Language Essentials*, and *Classes and Objects in Java.*

Tasks admit more than one subtask decomposition. Between each task and its subtasks a node is interposed (*R1* and *R2* in the figure) for each alternative ramification. These objects represent the adaptive element by which the structure of the course is made dependent on students' characteristics and their activity during runtime (see [4]). Their role consists of controlling which decomposition will apply at execution time when, as in the example, more than one alternative has been set. The activation of one or the other is determined by a predicate that is associated to each branching-node. These conditions are edited by selecting and dragging attributes from the user model onto a dialogue box where conditions are edited. This dialogue box is opened by double-clicking on the icon that represents the conditional branching-node (see figure 3). ATLAS allows combining simple comparison predicates between student properties, task parameters and/or literal values, in normal disjunctive form.



R2 activation: FALSE

	Student.Personal_characteristics.available_time	=	low
AND	Student.Programming_skills.OOP	=	yes
AND	Student.Programming_skills.other	=	yes
OR			
	Student.Programming_skills.C++	=	yes
AND	Student.Programming_skills.other	=	yes

R1 activation: TRUE

	Student.Programming_skills.OOP	=	no
AND	Student.Programming_skills.other	=	yes
OR			
	Student.Personal_characteristics.available_time	=	high

Figure 3. Constructing activation conditions

To make it simpler for the designer to deal with adaptive aspects, ATLAS allows working on specific user profiles provided by the designer, so that the tool infers generalised task models from partial specifications built for particular cases. To do this, the designer first edits the student model, from which the structure of tasks depends, adding new properties or removing them, and introducing values. From the examples provided this way, ATLAS instantiates the task hierarchy for the described profile, allowing the designer to directly edit the resulting structure, instead of the original representation, more abstract and hard to understand. In figure 2 this option is turned off. When "Evaluate Rules" mode is turned on in the toolbar, branching *R2* disappears for task *Java Programming*, because its activation condition is false (figure 3) for the provided student model attribute values. When the designer works in this mode, the system takes care of generalising the design and maintaining consistency.

ATLAS uses visual feedback to show the relations between the different model elements. Task to subtask, task to contents, and branching-node to student attribute connections appear as arcs between objects on the screen. In a realistic course with hundreds of interrelated tasks, this can become an extremely messy and unwieldy structure. To alleviate this problem and facilitate the navigation over large models under construction, ATLAS allows authors to select different levels of detail for different parts of the model. For example, connection arcs can be shown or hidden on a per-task basis. The amount of information displayed in each object can vary (option "Short View of Tasks"). The designer can also collapse or expand entire task subtrees with a single click. For instance in figure 2, the folder icon located at the lower right corner of task *Java Programming* indicates that the task is expanded, while *Language Essentials* and *Classes and Objects* do not show their subtasks. The remaining two tasks are not composite, so they do not have this icon.

The output from ATLAS is a model of the course ready to be processed by TANGOW, which provides runtime support. TANGOW generates web pages on the fly by executing branching conditions, assembling content fragments associated to tasks, and updating task state in response to student actions.

4. CONCLUSIONS

The development of high-quality educational products requires the direct involvement of professional educators in the elaboration of educational material. Beyond the selection or elaboration of text and multimedia resources, it is important for this involvement to have an effect on the definition of aspects like structure and functional characteristics of courses, as well as the interaction between the software and the student. We believe that the development of interactive tools that expressly support these aspects will contribute to the advance in this direction, allowing designers to spend more time and attention on pedagogical aspects, and less in solving technical problems.

The adaptivity of constructed applications allows for a wider reach for courses, favouring their diffusion over the net. Being aware of students' characteristics and the way they work is a key element in education quality, be it computer-supported or not. The description and explicit use of user models in a graphical environment to define adaptive capabilities in the constructed applications is a novel aspect both in the educational technology field and in user interface development.

There are many possibilities for continuation of the presented work. For instance, course contents are currently elaborated using HTML editors that are external to the system. Our plans for the near future include the development of a tool for content authoring that supports adaptive presentation. We are also planning to incorporate into the system other models that should influence course structure and presentation, such as the platform, context of usage, or the characteristics of the data handled by the application. Given the generality of the developed techniques, we will also consider their application to other domains with similar problems and needs to the ones we have been dealing with so far.

ATLAS is being developed in Java, JDK 1.2. A version of the tool is currently available at http://astreo.ii.uam.es/~atlas.

ACKNOWLEDGEMENTS

The work reported in this paper is being supported by the Spanish Interdepartmental Commission of Science and Technology (CICYT), project number TEL 1999-0181.

REFERENCES

1. M.W. Goldberg and S. Salari. *An update on WebCT (World-Wide-Web Course Tools)- a Tool for the Creation of Sophiticated Web-based Learning Environment.* Proceedings NAUWeb'97 - Current Practices in Web-based Course Development. Flagstaff, Arizona, 1997.
2. J. Vassileva. *Dynamic Courseware Generation on the WWW.* Proceedings 8th World Conference of the AIED Society. Kobe, Japan, 1997, pp. 498-505.
3. G. Weber and M. Specht. *User modeling and Adaptive Navigation Support in WWW-based Tutoring Systems.* Proceedings 6th International Conference on User Modeling (UM97). Sardinia, Italy, 1997.
4. R.M. Carro, E. Pulido, P. Rodríguez. *Dynamic generation of adaptive Internet-based courses.* Journal of Network and Computer Applications, v. 22, 1999, pp. 249-257.
5. N. Sukaviriya, J. Foley, T. Griffith. *A Second Generation User Interface Design Environment: the Model and the Run-Time Architecture.* Proceedings ACM International Conference on Computer-Human Interaction (InterCHI'93). Amsterdam, 1993, pp. 375-382.
6. P. Szekely, P. Sukaviriya, P. Castells, J. Muthukumarasamy and E. Salcher. *Declarative Interface Models for User Interface Construction: The Mastermind Approach.* In "Engineering for Human-Computer Interaction", L. Bass and C. Unger, eds. Chapman & Hall, 1996, pp. 120-150.
7. A. R. Puerta. *A Model-Based Interface Development Environment.* IEEE Software, v. 14, n° 4, 1997, pp. 40-47.
8. A. Cypher, ed. *Watch What I Do: Programming by Demonstration.* The MIT Press, 1993.
9. P. Castells and P. Szekely. *Presentation Models by Example.* In "Design, Specification and Verification of Interactive Systems '99", D.J. Duke and A. Puerta, eds. Springer-Verlag, Viena, 1999, pp. 100-116.

Guided Collaborative Chess Tutoring through Game History Analysis

Miguel Angel Mora, Roberto Moriyón
Departamento de Ingeniería Informática.
Universidad Autónoma de Madrid – 28049 Madrid, Spain.
Miguel.Mora@ii. uam.es
Roberto.Moriyon@uam.es

Key words: Tutoring, CSCL, Analysis, Chess, CSCW

Abstract: In this paper we introduce ChessEdu, a collaborative application that allows several people connected to the Internet to practise and learn chess in a guided setting. The main feature of ChessEdu is the use of representations of chess game histories, analysis and related annotations, which enhance the guidance of the teaching process. ChessEdu game histories play an important role from the didactic point of view, since they allow tutors to review what students have done and to guide their work. Annotations to the histories also allow tutors to include links to simplified positions. In this way, students learn chess by working on simple examples related to their previous wrong decisions in their game. ChessEdu is the first step in the definition of a framework to simplify the development of guided collaborative tutoring applications in a much broader context.

1. INTRODUCTION

During the last years there have been a number of experiments with collaborative applications that enhance the learning process [1, 2]. Tutoring systems that allow the guided collaboration of pupils among them or with a tutor can increase both the speed at which students learn the tasks and concepts they are supposed to master, and the deepness they reach in their knowledge. The development of telecommunications and multimedia technologies has increased the ability to build such systems up to a level that was hard to think not long ago.

However, traditional collaboration systems present some limitations about the role played by a tutor that in some cases can be crucial when using such a system in real life. The main limitation comes from the fact that a tutor who is in charge of several students cannot be paying attention simultaneously to the work all of them are doing. A system that helps teachers to spend their time guiding those students that need more support at each moment, and to review afterwards what other students have being doing and start another guided collaborative tutoring session with some of them, giving them guidance based on their actions, can be of great help.

In this paper we introduce ChessEdu, a collaborative application that allows several people connected through a computer network such as the Internet, including a tutor, to participate simultaneously or asynchronously in a chess game. The participants can play chess forming groups that share comments or analysis of different alternatives jointly; other participants can be synchronous observers of a game, or asynchronous analysts, in which case they can analyse the game in groups, synchronously, or by themselves, asynchronously, in which case they can communicate their ideas by adding alternative proposals to the development of the game or by making annotations that can be associated to some points of the game. ChessEdu can be used by students of any level, and by players who want to share their analysis of games synchronously or asynchronously while they are in different places.

Tutors can guide the players directly, moving the pieces of the board or making them synchronously review alternatives to the moves they made, or indirectly, creating annotations that can include links to stored games that illustrate similar situations. Also, they can review the analysis work of the students, reading their annotations, or joining the analysis session. In order to facilitate the tutor's work, the system provides agents that inform the tutor when a student or a group of students need help.

The system supports the participants switching from one type of collaboration to another. For example, it allows a player to analyse the history of the game s/he is playing between two moves. The work done by all the groups of collaborators about a particular game is integrated in the same history, producing a tree of chess moves, with their corresponding annotations. The use of tree-like histories enhances the asynchronous aspects of the collaboration, since it permits a tutor to review all the interaction that takes place between the different students and groups.

ChessEdu is part of the EnCiTec project that includes among its much broader goals the development of infrastructure for the development of collaborative aids to guided computer assisted tutoring. The last goal of this part of the project is to provide a generic framework for the development of collaborative applications that allow the collaborative analysis of the collaboration, both in a synchronous and asynchronous way.

The rest of this paper is organised as follows: first, we explore some previous work in this area, some CSCL systems and frameworks. Later, we will show some requirements to accomplish our main goals, using chess teaching as background, and how we have improved these, describing the chess tutor and the first prototype of the framework to be developed. Finally, we will expose some conclusions.

2. PREVIOUS WORK

Collaboration histories keep the information required for a later analysis of the collaboration among users. These histories can consist of video and audio data, or they can include a sequence of stored events produced by the application, that later can be reproduced, with the same application or with a different one. This kind of history can be used in asynchronous distance education, for example, to reproduce the resolution of exercises by students or to replay a lecture given by a tutor [3].

The analysis of collaboration histories has been used previously to evaluate the social level structures and processes that take place in a CSCL environment [4]. Also, there are some tools, such as CEVA [5], that support synchronous and asynchronous collaborative analysis; in this case the analysis is based on a history that consist mainly of video data.

One of the problems that appears when trying to incorporate an analysis of the histories is how to capture and reproduce the information produced through the normal collaboration session in a suitable way. A *video history* is not appropriate because it does not allow incremental history analysis. In order to accomplish this task we use the standard mechanism of event capture/replay present in frameworks for the development of collaborative applications, such as Habanero [6] or Disciple [7]. Later, the work of the users can be reviewed simulating these events, which reproduce the user actions.

3. REQUIREMENTS FOR GUIDED COLLABORATIVE TEACHING

In order to allow tutors to guide students synchronously by reviewing with them their previous work, guided collaborative teaching applications must give support for the use and management of histories of collaboration. This also allows the teacher to analyse at any given moment the work of the students, adding comments or alternatives to be sent to them, who can play them as animations.

The representation of collaboration histories by means of a tree maximises the flexibility of the analysis. This allows the system to keep track of different alternatives that are proposed.

The management of collaboration histories should include the support of collaborative Undo or Redo. With this, the teacher can show a better solution to the students, going backward in the history if necessary, and producing a new branch in it.

Other requirements of this kind of applications are:

1. Teacher and students play different roles in the collaboration.
2. Multimedia annotations, including synchronised voice and hypertext that can include references that in turn give access to states in the working process or in its analysis.
3. Dynamic collaboration sessions allow groups of users to split and do their own analysis following different paths.

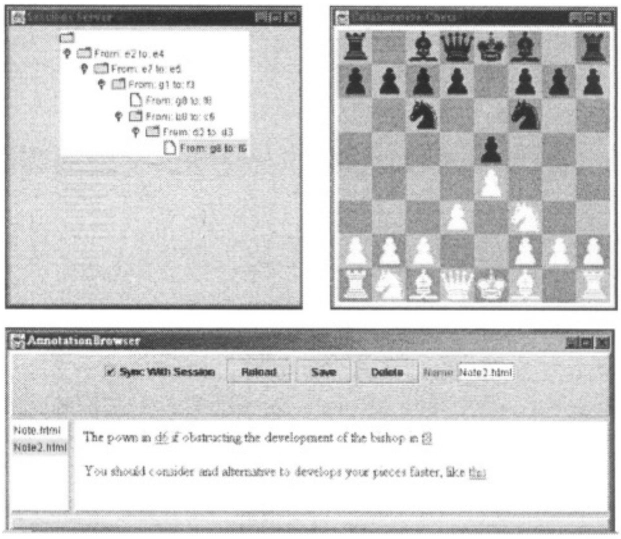
4. CHESSEDU

ChessEdu is a collaborative application that allows several people connected through a computer network like the Internet, including a tutor, to participate simultaneously or asynchronously in a chess game in any of the following ways:

1. Part of the people who are working collaboratively can be the players; on each side there can be just one player or a team. When more than one person play on one side, they can share their comments, and they can also analyse together different alternatives and make proposals before making any definite moves. Both comments and alternatives to be analysed include multimedia components, like voice and animations of parts of chess games. Actually, any game analysis can become an animation that can be included as part of the analysis of any other game. A tutor can be a player, in which case he/she has privileges to guide the members of his/her team by enforcing them to know his/her proposals and comments

and to accept his/her decisions with respect to the next move to be played.

2. Other people can be observers. A tutor can also be an observer; in this case, he/she can add comments to the history of the game to be reviewed later, or he/she can just select some positions he/she wants to go back to at some other moment.



The pown in d6 if obstructing the development of the bishop in f8

You should consider and alternative to develops your pieces faster, like this

Figure 1. A sample ChessEdu session

3. Other groups of people can analyse the history of the game asynchronously from the players, adding alternatives and comments to it. They in turn can be synchronised among one another or not. A tutor can also make asynchronous analysis of the game, either in isolation, in which case he/she can later have a reviewing session of his/her analysis with the students in order to guide them for future work, or synchronously with other participants in the session, who can be the players and/or other people; in this case, he/she also has privileges that allow him/her to lead the process.
4. Participants can switch seamlessly between being synchronised with the game or working on the analysis of its history. In particular, a tutor can interrupt a game, make the players go back to some point in the history with or without a lateral analysis and corresponding annotations, and send them back to work without any interruption.
5. When a participant plays the role of a tutor he/she gets reports from agents that analyse the different games, and calls his/her attention about

special situations that might need to be addressed directly between him/her and the students.

There are essentially three types of activities that participants can do during a ChessEdu session. These actions modify a tree whose nodes are moves in a game or in an alternative that is proposed to its development. The types of activities are: playing (i.e., adding new moves at the end of the game or an alternative for it), creating alternatives (i.e., adding new moves that start a branch of the collaboration history tree), and exploring the game and its alternatives.

ChessEdu is a Java application that extensively uses the rmi package of the Java 1.2 API. In order to use ChessEdu, a ChessEdu Session Server must be running in one machine. Each session represents a group of synchronous participants that work on a specific history or game. Several sessions can deal simultaneously with the same history or game; this is typically what happens when a tutor is revising the way the students have played and adding remarks and alternatives to their decisions.

ChessEdu (Fig. 1) supports two working strategies for the students: in the group learning strategy, two teams of students play against each other, and ChessEdu takes note in its multimedia logging of the proposals of each student; in this way, students get all the advantages of the classical group learning techniques, but the teacher has the advantage that he/she can take into account the students actions and correct the most important mistakes that he//she finds out among them. In the individual learning strategy either two students play against each other or one of them plays against the computer. The advantage of this strategy is that it enforces students to act, but on the other hand they get less stimulus from the environment.

Independently of which of these two strategies is used, ChessEdu gives bigger flexibility in the chess teaching process in two main aspects: first, the students as a whole can get a more direct and deeper idea of what they have done in the wrong way or just what they can do better. In the second place, they can also get more easily an idea about how they can do their tasks in a better way. This second dimension is enhanced especially by their possibility to work on examples of simpler situations where similar problems arise, and practice with them, either by themselves or synchronously with the teacher.

5. THE FRAMEWORK

A first prototype of a framework for the management of collaboration histories is .being developed in close relation to the development of the ChessEdu application.

Our present version of the framework is written in Java, and the applications communicate with the different servers in a transparent way by means of RMI, [8]. The framework can be used in the development of applications as well as applets, which can be started in a browser through the web.

In order to create a collaborative application with our framework, the following conditions must be satisfied:

- Events must be communicated to the server of the session.
- A method must be provided in order to listen to and reproduce events from other participants. Both regular and redo events must be handled.
- A similar method has to be provided in order to listen to undo events.
- The events that are transmitted must include enough information to allow the performance of both undo and redo actions.

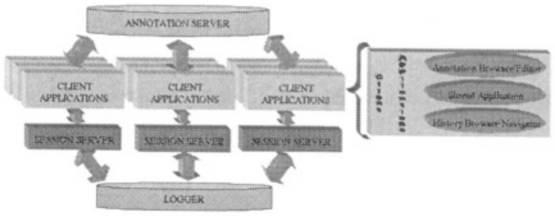


Figure 2. Framework Architecture

The framework gives the following services (Fig. 2):

- A Logger takes care of the storage of events produced by each session in a tree that represents a collaboration history. Annotations are attached to nodes in the tree.
- Sessions are connected to a logger, and take care of synchronisation among the collaborators that are associated to it.
- A History Browser and navigator allows the exploration and guidance of the collaboration history.
- The Annotations Editor allows users to read and edit HTML annotations associated to the nodes in the history, as well as the special links to other history nodes.

Currently, a first version of the framework that includes the Logger, Sessions, and History Browser services, as well as a prototype implementation of an Annotations Editor is implemented.

6. CONCLUSIONS

The usual way chess is learned by playing and analysing simple but representative positions collaboratively is very suitable for the implementation of a guided collaborative learning application.

We have shown how this kind of application can be enhanced by the use of game histories. This enhancement includes as an important feature from the didactic point of view the ability for a tutor to point out to the students their bad moves, their consequences, and related simple positions they should look at in order to correct their mistakes, while the benefits obtained traditionally in group learning are kept. Moreover, by the use of an application like this, tutors are able to pay attention to a bigger number of students with essentially the same degree of effectiveness. An application that satisfies all these requirements, *ChessEdu,* has been developed as part of the EnCiTec project.

We are developing a framework based on these ideas for the development of applications for guided collaborative tutoring.

ACKNOWLEDGEMENTS

This work has been funded by the National Research Plan from Spain through the EnCiTec project, project number TEL-99-0181.

REFERENCES

1. L. Lipponen, K. Hakkarainen (1997), "Developing Culture of Inquiry in Computer-Supported Collaborative Learning", CSCL'97 Proceedings
2. R. B. Kozma, (1999), "Students collaborating with computer models and physical experiments", CSCL'99 Proceedings, pp. 314-322
3. P. Thomas, L. Carswell, J. Emms, M. Petre, B. Poniatowska, B. Price, (1996), "Distance Education over the Internet", Proceedings of the conference on Integrating technology into computer science education, ACM press, pp. 147-149
4. K. Nurmela, R. Lehtinen, T. Palonen, (1999), " Evaluating CSCL Log Files by Social Network Analysis", CSCL'99 Proceedings, pp. 434-442.
5. Cockburn, T. Dale. (1997) "CEVA: A Tool for Collaborative Video Analysis", Group'97 Proceedings, ACM press, pp. 47-55
6. Chabert, E. Grossman, L. Jackson, S. Pietrowicz, C. Seguin (19989, "Java object-sharing in Habanero", Communications of the ACM, pp. 69-76.
7. W. Li, W. Wang, I. Marsic, (1999), "Collaboration Transparency in the DISCIPLE Framework", Group'99 Proceedings, ACM press, pp. 326-335.
8. RMI API, Sun Microsystems (1998): "Java 1.2 and JavaScript for C and C++ Programmers", John Wiley & Sons.

Using Analysis, Design and Development of Hypermedia Applications in the Educational Domain

A. Navarro, A. Fernández-Valmayor, B. Fernández-Manjon, J.L. Sierra
Dpto. Sistemas Informaticos y Programacion. Escuela Superior de Informatica, Universidad Complutense de Madrid. Madrid – Spain: {anavarro, alfredo, balta, jlsierra } @sip.ucm.es

Key words: analysis, design, development, hypermedia, education, XML

Abstract: In this paper we present ADDHA, a methodology for the Analysis, Design and Development of Hypermedia Applications. The development techniques entailed by this methodology can be applied to a broad spectrum of hypermedia applications. In this paper, we will apply them to the educational domain. We are interested in proving its viability in this area, one of the most demanding fields of application.

1. INTRODUCTION

Computers can be a valuable tool for the educational effort. Today we have computers in every educational institution, where they provide new additional facilities to students' education. The pervasive use of computers is opening new frontiers in the educational arena: online tutorials, specific databases and hypermedia applications are changing teaching strategies and creating new forms of interaction between students and teachers. Finally, we have the World Wide Web as the universal medium where all these activities can take place.

Because of its versatility, educational hypermedia applications are one of the most promising tools in the educational field. These applications can split its educational content accordingly to the natural structure of knowledge and to the factual relationships between conceptual items. This scheme can facilitate learner training through concept linking, and at the same time, can provide students with compelling interactions and feedback

[1]. However, in practice, when we design an educational hypermedia, the learner bounds are limited by the hypertext system [2]. These bounds are imposed by the educational content that underlies the application.

In this context, we can conceive the World Wide Web as a giant-like hypermedia system built over the linking of thousands of heterogeneous pieces of information where bounds are unattainable in practice, if not disappearing. Trying to organise all these pieces of information can be an endless (and perhaps futile) task, but trying to organise each individual piece is the task of every author that publishes in the web. Where the main problem that authors must confront when doing their job is the lack of structuring power of HTML [3], the standard language of the World Wide Web.

In this paper we present a methodology for the Analysis, Design, and Development of Hypermedia Applications (ADDHA). This methodology takes advantage of the structuring power of XML in order to build such applications. ADDHA is a generic methodology but it fits properly in the educational field. In the analysis and design phases, XML is used to make the structure of the educational content and the presentational structure of the application explicit in terms familiar to the customers. At the final phase of development, we can tune our design for various learning environments such as a stand-alone CD-ROM or a World Wide Web site. The techniques entailed by the methodology we present here are the result of our experience in the development of the educational hypermedia Galatea [4], a set of multimedia and hypermedia tutorials for the written and oral comprehension of Romance Languages.

The structure of the paper is as follows. In the next section we will describe ADDHA in an abstract mode. Then we will give details of how this methodology can be applied to the production of educational material. After that, we will provide the details of a brief application example. Finally, we will discuss some results and will present some conclusions and future work.

2. ADDHA LIFE CYCLE

Hypermedia applications development can be considered as a particular case of software development. Several models of software development process have been proposed to solve the chaotic nature of software construction. [5]. The main goal of these models is to provide developers with a set of well defined and repeatable steps to guide the construction of quality software. Nowadays the evolutionary model is the most commonly accepted. In this model, the life cycle is conceived as iterating over a number of activities, obtaining a prototype at the end of each iteration. In particular, we can depict this approach iterating over the classic waterfall model. Figure 1 shows a schema of this model.

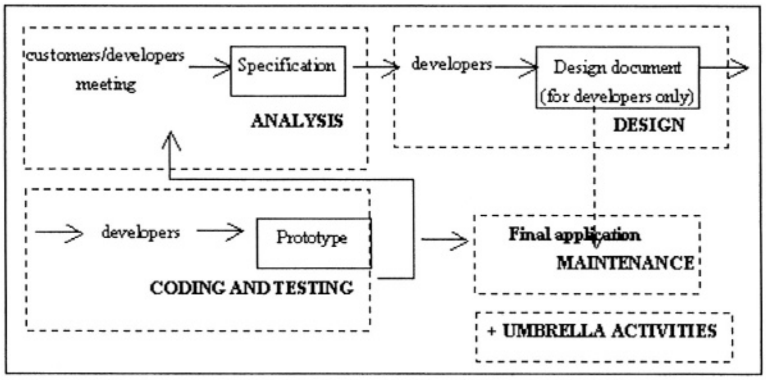


Figure 1. Iterating waterfall model

In this model customers provide an initial specification of the application that describes *what* the application has to do. This specification is the main product of the analysis phase. During the design phase, developers use this specification to decide *how* to build the application. The main product of this phase is a design document. This document must be a representation of the application, powerful enough to guide code development. After design, coding and testing follows. These phases can be iterated; at the end of each iteration we have prototypes that can be evaluated by the customers. The iteration process ends when the application meets customer requirements. Next, we have the maintenance phase, and in parallel, there are several umbrella activities whose goals are to assure the integrity of the process.

In the design and development of hypermedia applications, customers play a central role, providing not only the content and the usual specifications, but also the far more subtle specifications of the anticipated user interactions [6]. To confront this problem we propose a modification of the previous model trying to involve users in the software life cycle. The main issue is to find a formalism precise enough to be useful for developers, and simple enough to be useful for customers. A deeper study of the relationships between educators (customers) and developers can be found in [6]. Figure 2 depicts our approach.

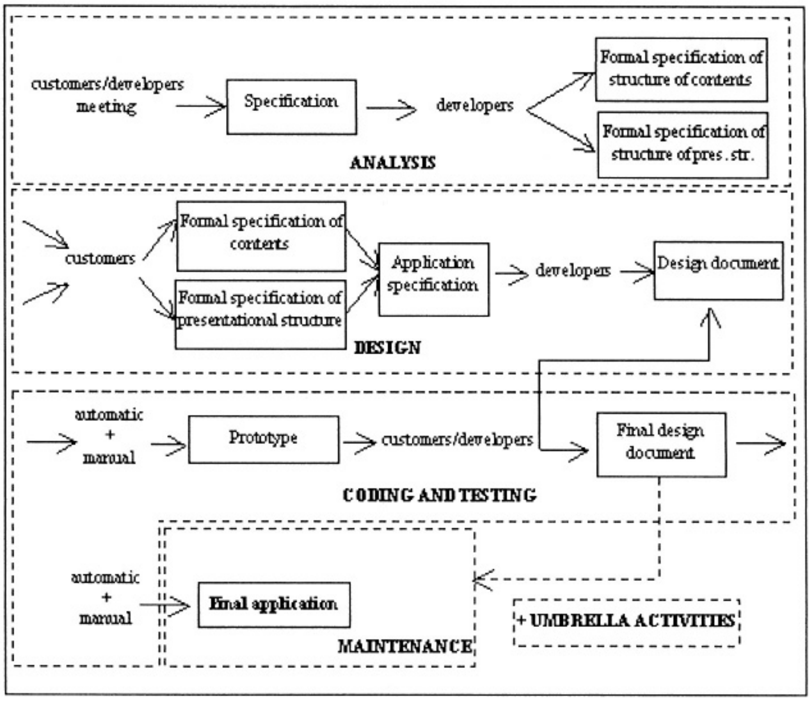

customers/developers meeting
Specification
developers
Formal specification of structure of contents
Formal specification of structure of pres. str.
ANALYSIS
customers
Formal specification of contents
Formal specification of presentational structure
Application specification
developers
Design document
DESIGN
automatic + manual
Prototype
customers/developers
Final design document
CODING AND TESTING
automatic + manual
Final application
MAINTENANCE
+ UMBRELLA ACTIVITIES

Figure 2. ADDHA life cycle for hypermedia

In ADDHA we use a divide and conquer approach applied to software life cycle. During the analysis phase customers and developers produce the specification of the application. Using this specification, developers provide formal specifications of the structure of the informational content (knowledge), and the anticipated user interactions (mainly the presentational structure). At the next phase, customers will use these specifications as templates to provide the actual contents and presentational structure of the application. These templates are then combined to provide a first specification of the application. This first specification can be incomplete, probably by two reasons. First, it could be difficult for customers to specify the whole application using formal templates. In this case customers are free to use natural language to complete the specification. Second, we must consider the inclusion of subordinate processes in the application. A subordinate process is a small program that, running inside the application, will normally be activated by some user action (for example, users can ask the system to evaluate their answers to some exercise). This specification is completed by developers including design diagrams that represent the subordinate processes. The completed specification constitutes the *design document.* Developers use this document to build the prototypes of the application. When developers and customers discuss over the modifications of the prototypes, they can use the design document to reshape the prototypes, facilitating their interaction. Once the application has been completed, the design document is used as a guide to support maintenance. The umbrella activities are the common ones, and they do not substantially change in ADDHA.

3. ADDHA AT WORK

In the previous section we have outlined the ADDHA methodology in an abstract manner. Now we are going to describe the previous abstract steps of the ADDHA life cycle in terms of specific developing techniques. Figure 3 shows the changes.

In ADDHA, our divide and conquer approach must produce different types of documents, each of them describing a different facet of the application. We use XML DTDs [7] to formalise these different types of documents and to provide educators with the templates they must work with. The formal specification of the structure of the content is provided by a DTD called *content-DTD*, and the presentational structure is provided by another XML DTD called *presentation-DTD*. The presentation-DTD is generic for families of hypermedia applications. But content-DTD is usually specific of each application. Using content-DTD, customers (educators in this case) structure the knowledge, i.e. the informational content, of the application. This stage is crucial and probably is the most critical step in the process of developing an educational hypermedia [4]. The result of this stage is the *content document*, that is, an instance of the content-DTD (in a moderately complex application, we will usually have more that one instance for a content-DTD and perhaps we will need more than one content-DTD to structure all the knowledge held by the application). The XML-elements of the presentation-DTD represent the presentational elements of the application and the relationships between them. Basically, these elements are windows (real or abstract channels of communication with the user) and interaction items such as buttons. In consequence, describing the presentational structure of the application is equivalent to building an instance of the presentation-DTD. In addition, the elements of presentation-DTD have assigned an object oriented semantics (in terms of the desired language object oriented construction, e.g. Java classes), so when you build the instance of the presentation-DTD you are describing the skeleton of the object oriented code that is going to implement the running application.

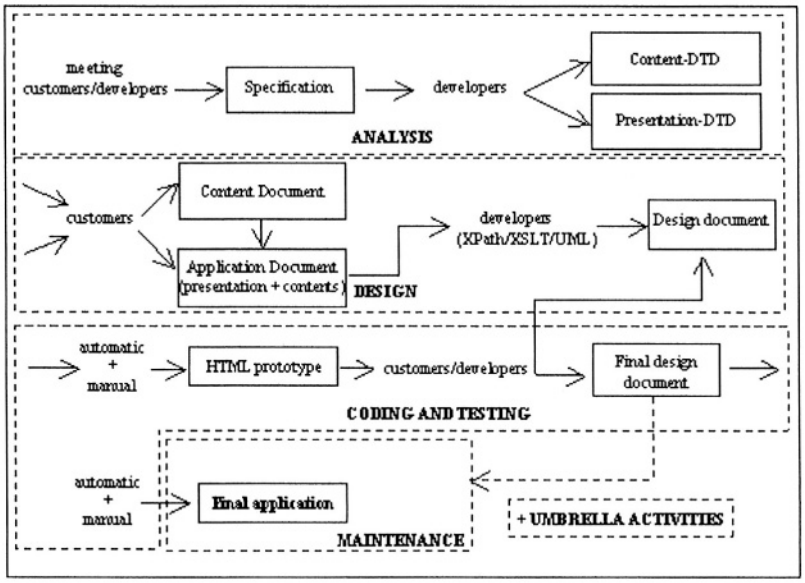


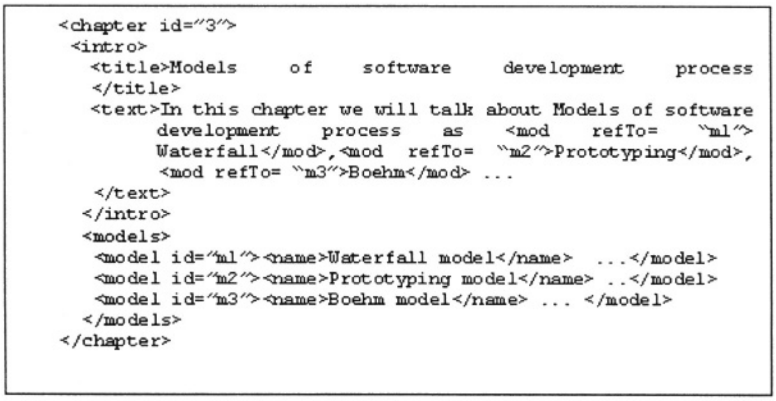
Figure 3. Concrete ADDHA life cycle for hypermedia applications

The construction of the application specification, conceived as the combination of the formal specifications of contents and presentational structure in the abstract ADDHA life cycle, is simplified in practice. Applying ADDHA, the specification of the presentational structure of the application, and the assignment of the contents to this structure is accomplished in the same step.

We have devised specific mechanisms to combine different types of documents (the instance of the content-DTD, and the instance of the presentation-DTD). One of these mechanisms is overmarkup and we implement it at the DTD level. The overmarkup idea is very simple. The presentational structure of the application (e.g. screens and windows) is described using the structure provided by the presentation-DTD. Therefore, to complete the total definition of the application we only have to describe which contents appear inside each element of that structure. In this context, the contents of the elements of the instance of the presentation-DTD are (mainly) parts of the content document (that is, the instance of the content-DTD). There are several ways to physically implement overmarkup. One is using a special binding that relates both instances (a sort of cut and paste). This is simple, but it is not very elegant. Other option is to use a language that can refers to parts of instances of XML documents, such as XPath [8] expressions. If we need to make any change in the content document, we can use any transformation language, for example XSLT [9]. The separation of contents and presentation provides several advantages, recognised in all hypermedia areas [10]. The clearer one is to be able to assign different presentation to the same contents, or vice versa, the same presentation to different contents. The overmarked document is called *application document,* and it is finished by developers when they provide the complete XPath/XSLT expressions, and object oriented diagrams [11, 12], obtaining the *design document.* As a consequence of the process, this document is partially built by customers, and so, easily understandable by them (educators in our case). The design document is also precise enough to provide a complete description of the hypermedia application that can be used by developers at the coding phase.

4. A BRIEF EXAMPLE

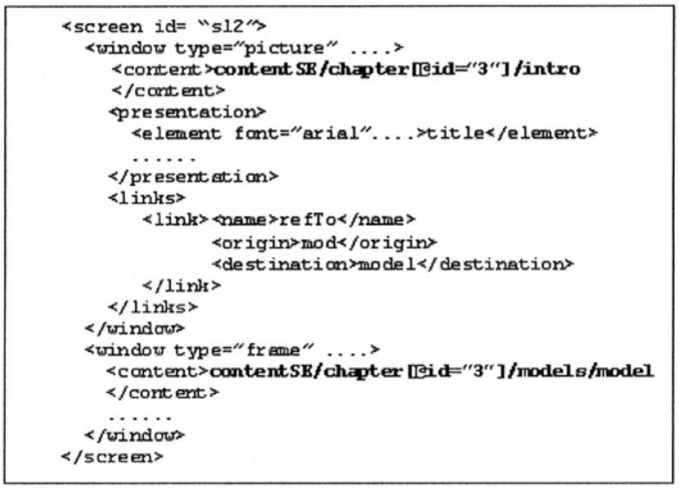
We are going to illustrate these ideas with a very basic example. Imagine that a team of university professors wants to make a set of simple educational tutorials on a selected subject. The application of ADDHA to this task would be as follows. First, developers meet every educator to agree about the structure of the contents of every tutorial (say tutorial 1, tutorial 2, ..., tutorial n). After this, developers provide a content-DTD for every tutorial (content-DTD 1, ..., content-DTD n). Then, educators and developers agree about the common presentational structure of the courses. This structure is formally described in the presentation-DTD. Every educator fills its application document using this DTD as a template.



```
<chapter id="3">
 <intro>
   <title>Models    of    software    development    process
   </title>
   <text>In this chapter we will talk about Models of software
         development   process   as   <mod   refTo=   "m1">
         Waterfall</mod>,<mod  refTo=  "m2">Prototyping</mod>,
         <mod refTo= "m3">Boehm</mod> ...
   </text>
  </intro>
  <models>
   <model id="m1"><name>Waterfall model</name>  ...</model>
   <model id="m2"><name>Prototyping model</name> ..</model>
   <model id="m3"><name>Boehm model</name> ... </model>
  </models>
</chapter>
```

Figure 4. Part of the content document

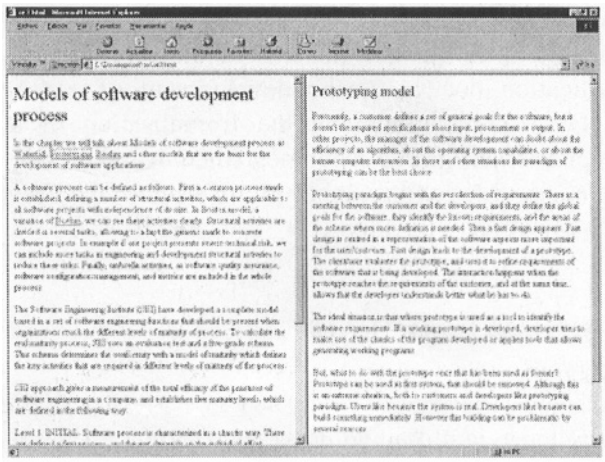
For example, suppose that one tutorial is centred on Software Engineering, and that there is a chapter about models of software development process. With independence of the particular presentation of each chapter, there is a well-defined structure for all of them, captured by a specific content-DTD. In this way, educators can specify the content of the hypermedia application, in a structured manner (using their specific vocabulary) with independence of the presentation of each content. In this example we have an introductory text with the name of the models that hyperlinks to its description. The content document for this chapter is described in Figure 4. Now it is possible to assign the presentational structure of each of them. For example, suppose that the introduction and the models descriptions appear in two different windows on the same screen. The application document that describes this presentational structure is shown in Figure 5. In this figure we can see a screen specification with two windows. In one appears the introduction (using the XPath expression `contentSE/chapter[@id="3"]/intro`), and in the other the selected models (again using another XPath expression).



```
<screen id= "s12">
  <window type="picture" ....>
    <content>contentSE/chapter[@id="3"]/intro
    </content>
    <presentation>
      <element font="arial"....>title</element>
      ......
    </presentation>
    <links>
       <link><name>refTo</name>
              <origin>mod</origin>
              <destination>model</destination>
       </link>
    </links>
  </window>
  <window type="frame" ....>
    <content>contentSE/chapter[@id="3"]/models/model
    </content>
    ......
  </window>
</screen>
```

Figure 5. Part of application document

The application/design document also contains valuable information to build the complete application. Figure 6 shows the basic HTML prototype automatically obtained from the application document. Note that an important part of the prototype can be built automatically from the application document, but the inclusion of subordinate processes (represented by object-oriented diagrams) would need manual coding.



Models of software development process

[illegible]

Prototyping model

[illegible]

Figure 6. HTML prototype

If you want to assign a different presentation to the same content (e.g. the description of the models will appear in a new window) only changes in application document are needed. Separating content from presentation includes advantages in both directions. If you want to change the content to include another model process, you can do it, changing the content document, without changing anything in the design document, because the XPath expression that points to the introduction and to the models does not change.

5. CONCLUSIONS AND FUTURE WORK

The development of hypermedia applications is basically a software development activity. This activity turns harder by the direct implication of customers providing contents and anticipated user interactions with the application.

In hypermedia applications with a very structured content, such as the educational ones, the need for structuring is a primary goal. ADDHA techniques fit properly with the construction of educational hypermedia applications providing structuring of contents, and facilitating the interaction between educators and developers via the application/design document. These techniques gather the benefits of separating the contents from the presentation, changing one of them without affecting the other. In addition, they permit the inclusion of object-oriented specifications for subordinate processes, and present a straightforward translation to object-oriented code due to the direct relationship between the elements of the presentation-DTD and the components of the object-oriented language.

Future work includes: the automatic generation of object-oriented code using the application document, the development of a CASE tool that facilitates ADDHA application, and the formalisation of a hypermedia model that supports ADDHA techniques (then we will have MADDHA techniques, that is, Modelling, Analysis, Design and Development of Hypermedia Applications).

ACKNOWLEDGEMENTS

The EU Galatea project (TM-LD-1995-1-FR89) and the Spanish Committee of Science and Technology (TIC97 2009-CE and TIC98-0733) have supported this work.

REFERENCES

1. Norman, D.A., Spohrer, J.C., *Learner-Centred Education.* Communications of the ACM, 39, 4, 24-27, 1996.
2. Leggett J.J., Schnase J.L., Kacmar C.J., *Hypertext for Learning,* NATO ASI Series 27-37, Springer Verlag, 1990
3. W3C, *Hypertext Markup Language (HTML) 4.01. Specification* http://www.w3.org/TR/html4/, 1999.
4. Fernandez-Valmayor, A., Lopez-Alonso, C., Sere, A., Fernandez-Manjon, B., *A hypermedia design for learning foreign language text comprehension.* Proceedings of IFIP WG3.2/WG3.6 August 1999 Working Conference on Building University Electronic Educational Environments, 1999.
5. Pressman R.S., *Software Engineering: A Practitioner's Approach,* McGraw-Hill, Inc., 1997.
6. Navarro A., Fernandez-Manjon B., Fernandez-Valmayor A., Sierra J.L., *A Practical Methodology for the Development of Educational Hypermedias*. Proceedings of ICEUT 2000, 16th IFIP World Computer Congress 2000, Information Processing Beyond Year 2000, in press, 2000.
7. W3C, *Extensible Markup Language XML Version 1.0.* http://www.w3.org/TR/REC-xml, 1998.
8. W3C, *XML Path Language (XPath) Version 1.0.* http://www.w3.org/TR/xpath, 1999.
9. W3C, *XSL Transformations (XSLT) Version 1.0.* http://www.w3.org/TR/xslt, 1999.
10. Halasz, F., Schwartz, M., *The Dexter Hypertext Reference Model.* Communications of the ACM, 37, 2, 30-39, 1994.
11. Booch G., *Object-oriented analysis and design with applications*, Second Edition, Benjamin Cummings Publishing Company, 1994.
12. Rumbaugh, J., Booch, G., Jacobson, I., *Unified Modelling Language Reference Manual.* Massachusetts: Addison-Wesley, 1998.

Ubiquitous Computing and Collaboration

New interaction paradigms in the classroom for the 21st Century

M.Ortega*, M.A.Redondo*, M.Paredes**, P.P.Sánchez-Villalón***, C.Bravo*, J.Bravo*

Grupo CHICO

** Escuela Superior de Informática, Paseo Universidad s/n, Universidad de Castilla - La Mancha 13071 Ciudad Real*

{mortega, mredondo, cbravo, jbravo}@inf-cr.uclm.es

*** Escuela Superior de Ciencias Experimentales y Tecnología, Universidad Rey Juan Carlos 28933 Móstoles, Madrid*

m.paredes@escet. urjc. es

**** Escuela Oficial de Idiomas, Paseo de la Universidad s/n, Ministerio de Educación y Ciencia 13071 Ciudad Real*

ppsanch@fimo-cr.uclm.es

Key words: collaboration, ubiquitous computing, e-learning

Abstract: In this paper we are presenting the foundations and groundwork of the activities performed by the CHICO team at UCLM in order to study the methods and techniques to design the so-called Classroom for the XXI Century. Our assumption is to establish the conditions to develop the classroom in which the principles of ubiquitous computing and collaboration will meet together to accomplish and help CAI objectives.

1. INTRODUCTION

The worldwide use of computers in all the fields of human activity has grown to tremendous popularity over the last few years. We cannot claim so about their use in the education domain, where the introduction of computers, although still being gradual, does not reach the usage range and acceptance necessary to make the desired qualitative leap forward for education. The simple fact of placing a computer in the classroom does not guarantee its use. However, there are some data [1] that confirm that everyday lessons (in Higher Education in the USA) have been undergoing the most significant change in many decades, a change directly influenced by the use of computers in the classroom. The members of the CHICO team (Computer Human Interaction and Collaboration) are conducting an ambitious project centred on the experimentation and research of new interactive technologies susceptible to be applied in the classroom. Thus, we try to change the traditional concept of a lesson or class. In particular, studies and developments are being carried out to turn the classroom into a ubiquitous computing environment where computer-assisted collaboration help students to solve complex problems and develop advanced skills.

To a certain extent, the personal computer has been introduced in the classroom, but its use is not coming up to initial expectations, as, indeed, it initially happened with audiovisual aids and other similar tools. The provision of computers in school labs, the attempts to develop a large amount of computer material and software for the subjects to study, the training in their use and the initial motivation are not bringing about their generalised use. It seems as if there were primary, basic elements in the educational activity in the classroom hindering the progress, impeding the immediate change to another tool that can theoretically accomplish the same or better results. There are social habits necessary to evolve but impossible to replace immediately. We are referring, for example, to the use of such elements as the blackboard, the paper and the pencil. Ubiquitous computing will allow their evolution, without replacing them. In this article we present the groundwork and the architecture of the first implementations undertaken. In section 2 we introduce the concept of ubiquitous interaction and the hardware necessary to make it possible, according to Mark Weiser's view [2], applying the current technical developments and our view of ubiquitous computing. In section 3 some principles are described on which lay our ideas of collaborative environments. Section 4 describes the model established for the first implementations. In particular, a language learning activity case as an instance for a ubiquitous class, carried out as a real experiment. Finally, section 5 determines some conclusions that have already been reached and the directions to guide our work in the future.

2. UBIQUITOUS COMPUTING

Ubiquitous computing as the interaction paradigm was first introduced by Mark Weiser in Xerox PARC laboratories in Palo Alto in 1991 [2]. This new paradigm changes the concept of using the computer by distributing multiple low-powered computers or small computer components throughout the environment, trying to hide their presence, concealing their use [3] by disguising their appearance adapting them to the traditional tools used in the classroom. In other words, ubiquitous computing tries to extend the computational capacity to the whole environment by means of the distribution of small and very diverse devices with specific interactive characteristics, connected to higher-powered servers. The design and location of these devices must be thoroughly analysed according to the task they are to perform. This way, the computational responsibility is dispersed, loosely distributed in the environment, giving an impression of omnipresence [4].

In his already well-known article [2] and in some later works [5], Weiser described with detail his revolutionary views: how computer ubiquitousness can help us in our lives, without being intrusive to us. The solution proposed by this author consists in having wireless networks of computers interchange information between each other and serve as interaction mechanisms. In ubiquitous environments there are three types of computer devices: "badges" or "tabs", "pads" or "tablets" and "whiteboards".

The *badges* are small machines, a few centimetres wide and long, similar to labels but unlike these they are active. They have a small microprocessor and infrared technology to transmit the identity and the location of the person bearing them. These data can make doors open just by approaching them, or similarly, adapt some information displayed on a screen to the badge holder. The *tablets*, as large as a LCD screen, are used in the same way as we use the paper, a book or a magazine. They have a radio frequency connection with enough channels to link hundreds of such devices, enabling communication to every person in a different room. The objective of these tablets is not for the users to permanently carry them but to use them where they are with no type of associated identity, a characteristic radically different from portable computers. Finally, the *whiteboards* have the habitual size of those in a classroom and serve as any device used to show information, i. e., as a video monitor, a notice board, a display panel, or even as a screen shared to download information coming from the badges and tablets or to upload data to these. In order to use them a marker, a piece of "wireless electronic chalk", is used.

These computers were designed at Xerox Research Center in Palo Alto [5], and in a later article [6] Weiser affirms that it is at the university where his idea of ubiquitous computing can have more acceptance. In 1999 Ortega [7] reckoned on the implementation of this new paradigm in the classroom with educational aims, presenting some of the benefits that its use will provide in education. Soloway [8] considers that handheld type or PDA devices mean a new incentive in the way to exploit the power of learning by discovery. Following the principles of the ubiquitous computing paradigm, He conceives these devices as remote controls of a group whiteboard. He proposes to design peripheral devices and use these, say, handheld devices, for data capturing in a real environment. Later these data will be sent to a server which will display them for group discussion. The dream of the ubiquitous computer is on the verge of becoming real with these tools. Devices with high calculation capacities and capable of interconnection without cables are a reality. As some authors [8] suggest, if we add the possibility of handheld devices interconnected by means of wireless networks we practically have the system described by Weiser in 1991.

3. COLLABORATIVE SYSTEMS WITH UBIQUITOUS COMPUTING SUPPORT

From the learning perspective, it is necessary to emphasise the theory of collaborative learning as the central source of support in our expositions. In the search of a theoretical framework to use as reference, we are to emphasise the CSCL paradigm (Computer Supported Collaborative Learning) [10]. This paradigm is mainly based on social constructivist theories [11], the theories on situated and shared cognition [12, 13, 14] and mainly on the Soviet historical-cultural theory arisen from the works by Vygotsky [9]. According to this author "In a collaborative scenario, the students exchange their ideas to coordinate in the attainment of shared objectives. When dilemmas arise at work, the combination of their activity with communication is what leads to learning". Many systems have been developed on the basis of the principles of the CSCL paradigm, to mention some: the CoVis project (Collaborative Visualisation) arid its "Collaborative Notebook" [15], CSILE [16], CaMILE [17], KIE [18], C-CHENE [19], SCOPE [20], etc. In the Spanish university we can point out the developments contributed by a group at UNED in DEGREE [21], and even, in all modesty, the CHICO team firmly reckon on these principles in DomoSim-TPC [22, 23, 24].

However, when we try to fuse the principles of the CSCL and those of ubiquitous computing we must consider certain aspects. In this sense we dare to establish a classification of systems according to the characteristics of ubiquity involved and to the kind of collaboration adopted:

– Systems that only support file downloading and uploading.

This type of systems implement mechanisms for downloading file from a server to mobile devices in order to get them work off-line later. Then the resulting work can be transferred to the server for storage. In order to approach the design of such a system we only need a PDA or a handheld device and some software tools based on file management, email or hypertextual interfaces.

– Systems that support collaboration in asynchronous discussion interfaces.

The aforementioned systems lack group work facilities. In order to progress over this obstacle there is another type of systems which, in addition to the information management functionality, provide a set of tools for discussion making it possible to present individual work to the group. These systems define both an individual workspace and a group workspace for discussion, even in some cases areas of results, but only of an asynchronous type.

– Systems that support synchronous collaboration.

The asynchronous discussion facility provides innumerable benefits from the social point of view. However, in a great number of situations and domains it becomes essential to have tools for interaction and communication in real time. The systems that support participation and discussion in real time are more complete but also more complex. In these, in addition to the definition of workspaces, it is necessary to define work sessions and protocols [25] in order to structure and lead the interaction of the participants with the system. We can mention some of the works of Soloway and its collaborators [8] as examples of this type of systems, still susceptible of improvement (mainly as to collaboration). The CHICO team commit their research work to systems supporting collaboration in real time and rely on ubiquitous computing as the new interaction paradigm to design the model that Ortega [7] claimed to be the classroom of the future, already feasible at present.

4. A MODEL OF AN ELECTRONIC CLASSROOM FOR LANGUAGE LEARNING

Our objective works on the definition of a classroom model based on the principles of ubiquitous computing and computer-assisted collaborative learning as a low cost system, to apply it to language teaching as a case study. Thus, we are going to describe the architecture proposed below (Figure 1). Then, we are going into the most important details about the application of this case study.

4.1 Architecture to support ubiquitous computing and collaboration in the classroom.

The architecture is based on the use of several wireless network technologies. We have used some commercial devices and others developed in our laboratory. It includes the following elements:

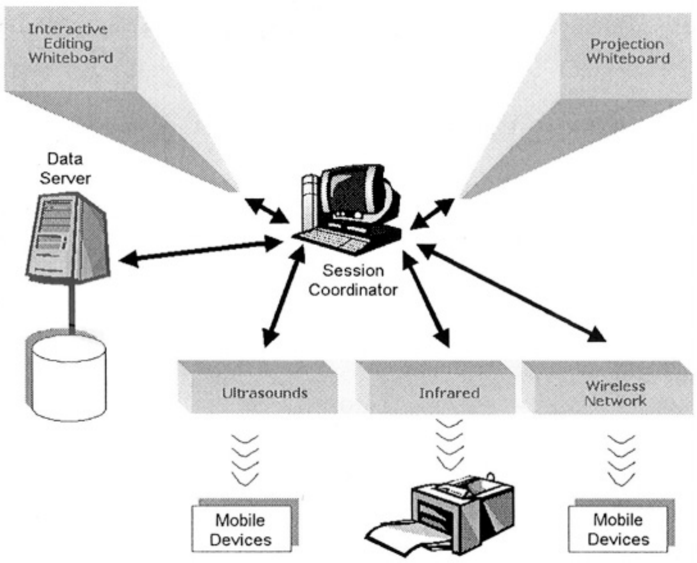


Figure 1. Model to implement ubiquitous computing and collaboration.

4.1.1 Wireless network

As wireless network we used a commercial product, ELSA AirLancer, a set of wireless connections instead of the standard wiring of a local area network (LAN). This brings about wireless local area networks (WLAN) that have the same functionality as a LAN. They improve these facilitating the integration of mobile devices. Wireless network cards use the IEEE 802.11b standard, a modification of IEEE 802.3 for Ethernet connections. IEEE 802.11b standard uses a range of ISM-type frequencies (Industrial, Scientific, Medical) from 2.4 to 2,483 GHz, with a bandwidth of 11 Mbps. They can reach a range up to 300 metres outdoors and around 30 metres inside buildings, with an average speed of 5.5 Mbps transference, which provides an excellent solution for the classroom environment.

4.1.2 Infrared transmission and reception.

The transference of information by infrared transmission is achieved by using devices with the IrDA standard (Infrared Data Association) [26], which was specially designed for communicating palmtop devices, laptops and mobile telephony terminals. At present, there are a lot of peripherals with this type of communication ports for wireless connection and communication. Even when there are no such connections we can expand the tool by means of IrDA devices that directly connect to the RS232 serial port.

4.1.3 Ultrasound devices.

In our model we also use ultrasound technology to determine the position of the mobile devices. They have an ultrasound emitting device to be located. They work the same way as Weiser's active badges. Commercial devices of this type exist, even at very low cost, used for tasks as simple as length measuring. However, we made use of previous works already developed in our university for mobile robotics [27]. They provide measures with a 2 cm margin for error.

4.1.4 The Whiteboard Projection Set

For whiteboard projection we use a whiteboard and an overhead projector (OHP) connected to the VGA card of the computer used as the session coordinator. This way, when required, it is possible to have the information projected on the whiteboard.

4.1.5 The Interactive Editing Whiteboard

In addition to the Whiteboard Projection Set (or as an extra tool attached to the Whiteboard), we use a device capable of capturing what is drawn or written with a marker (or electronic chalk) on the Whiteboard. Once caught, this information can be processed and be sent to all the mobile devices through the wireless network. This way everybody can share the information in real time. There are some research works in this line such as the ones described in [28] and the ones based on artificial vision [29]. Nevertheless, we have decided on a low cost commercial device called *mimio*™ of Virtual Ink™. It is an easy to use, portable hardware tool. It stores, reproduces and prints the information drawn on a whiteboard. It incorporates a bar-like sensor attached to one of the whiteboard sides. It is connected to a series of sensors to be placed in markers and to another device, the eraser, to clean or erase the whiteboard. The Editing Whiteboard constitutes itself a powerful tool to make presentations and even to share information through an Intranet or the Internet. The connection of the *mimio*™ to the session coordinating computer is made by means of a RS232 serial port, but, as seen above, everything seems to indicate that we can do it by means of an IrDA connection. Tests are being developed in this sense to reach this objective.

4.1.6 Session Coordinator

The Session Coordinator is a computer with communication interfaces to connect all the described technologies previously referred. Therefore, it is the one in charge of receiving, processing, storing and distributing all the information flow. To this purpose, another computer is in charge of the data management, doing the functions of an information server implemented in a database management system (in our first experiences we are using MySQL accessed through JDBC).

4.1.7 Mobile devices

The mobile devices, as Figure 2 shows, are formed by a set of PDA (Cassiopea of Casio or iPAQ of Compaq) using a PCMCIA card or MMCard extended module for integrating the hardware necessary to have access to the wireless network to enable information interchange. Additionally, they have an ultrasound transmitter/receiver device integrated so that this way the device bearer's location can be tracked in the environment. They can do the work of both badges and tablets, which enables us to implement both tools in just one device. Thus, the Session Coordinator can determine that a person with a mobile device is approaching the whiteboards and can, for example, directly project the information contained in their PDA on the Whiteboard Projector, reproducing the messages they bear, to mention one of its utilities.

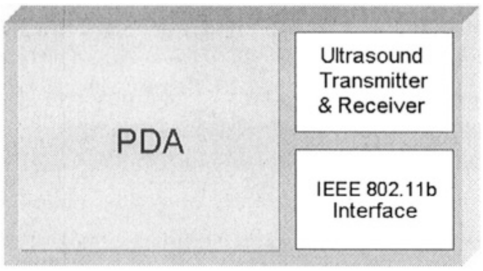


Figure 2. Mobile device.

4.2 The model applied to Language Learning.

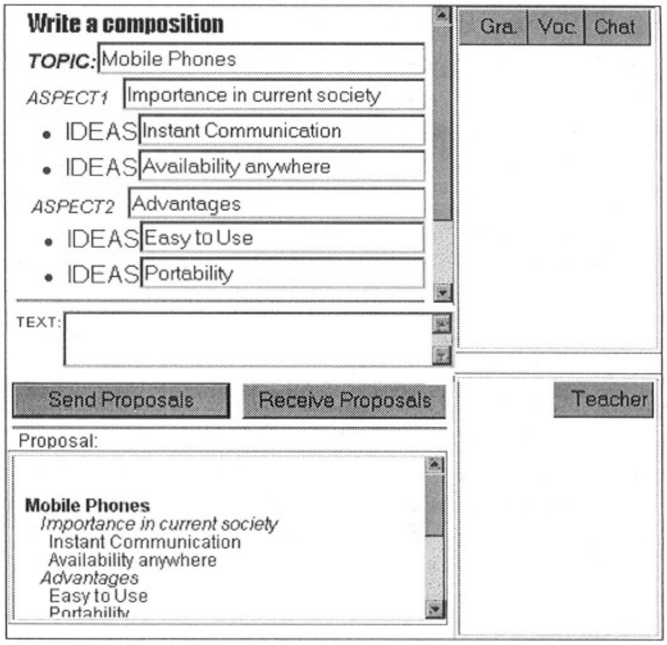
In order to implement this model we are presenting a traditional activity, the writing of a composition in class, in the domain of language learning, in particular, English as a Foreign Language (EFL). This case study already poses an innovation itself: to practise the writing of English in a collaborative environment. Here a group of students work together to get a common goal: to write an essay or a project in English. At the same time, all the process is guided by the computer that will as well participate in the composition development by means of the ubiquitous computing paradigm.

Generally, in the teaching of English, composition writing is an activity where the students have to write on their own. They are given, at most, some instructions on the type of composition to make, whether they have to write a description, an explanation, a narration, a letter, a report or the expression of their views, and the title or topic. Usually, the writing of a composition is either a homework task, with access to all type of aids available, or else a testing technique. In both cases it is an individual activity.

In our class we propose this task with the habitual instructions, but the students must work in group and at any moment they can get help from the computer, from the other members in the group and be guided by the teacher.

In writing a composition, as a well structured activity for language learning, the student is usually advised to put down a schema with the main aspects to write about. Brainstorming follows in order to develop these aspects. This is achieved by means of note making and it is an eminently conceptual or lexical process, made up of phrases mainly. Once those ideas have been organised and selected according to the relevance of their contribution to the development of the aspects, the students begin to extend them into correct sentences and coherent paragraphs. This is the syntactic part, the one of formation of the grammatically correct sentences and of the textual organisation of these into connected paragraphs.

This way, in a ubiquitous collaborative environment, the teacher tells the students the topic and the type of text to write. The students begin the activity with the process of brainstorming, suggesting aspects and ideas to the group by means of their mobile devices. They use them to send and to receive all the set of suggestions (collaborative brainstorming), which will be shown on the Whiteboard as well. For this purpose, we have developed an easy to use interface by means of a document in dynamic hypertext language DHTML. It consists of an interactive form where text boxes are available in order to insert the annotations and buttons are provided for their submission and reception. These activate the communication through the server (Figure 3).



Write a composition

TOPIC: Mobile Phones

ASPECT1 Importance in current society

- IDEAS Instant Communication
- IDEAS Availability anywhere

ASPECT2 Advantages

- IDEAS Easy to Use
- IDEAS Portability

TEXT:

Proposal:

Mobile Phones
Importance in current society
Instant Communication
Availability anywhere
Advantages
Easy to Use
Portability

Figure 3. Application Interface.

The students can receive the content proposals of the group and thus decide on which they consider most appropriate and excellent for the composition. At the same time, they can add, if they wish, other derived ideas or extend either the suggested ones with more precise or complex annotations. At this point, the first grammar elements soon begin to appear to cohere the notes (content words and phrases) into whole ideas (more extensive phrases and sentences). Next, the discussion process is carried out to select the most significant contributions and to organise them in the most appropriate order. In this process it is important to point out the role of the teacher as a guiding expert, controlling the level of knowledge and the adequacy of language use, suggesting an optimal plan with the proposed elements, etc. This can be carried out orally or with written messages enabled by means of a chat interface.

When the group reach an agreement on the selection and organisation of the notes, the process of text formation begins. Here appears the role of the computer as an aid. Depending on the level of language knowledge the group should have, the lexical choice can vary from very basic words, with frequent repetitions, to less frequent, richer or newer vocabulary which they can come to use without being sure they are correct. Our system facilitates access to dictionaries of all types: monolingual, for common definitions and word combinations ("collocations"), as well as bilingual, to translate new concepts, and thesauruses, to avoid repetition and to learn new vocabulary. They can also revise the resulting text by contrasting special expressions in a lexical database arranged for this purpose in the system. The lexical database is composed of a variety of original texts and a case library, i. e., the compositions previously made corrected by the teacher. We are implementing a programme for syntactic analysis to use as grammar corrector and style checker, but the student can already count on a grammar reference tool which gives help depending on the type of composition to write.

The teacher's role involves not only the correction of mistakes but also guidance as an expert as previously referred, and, after the first part of the activity, as an organiser in the distribution of tasks, making some students revise for grammar mistakes, others for vocabulary choice, and others for connectors, textual organisation and style of the composition. Once finished, the text is proposed for the group's general approval and for formal hand-in to the teacher, who evaluates it and includes it in the case library.

5. CONCLUDING REMARKS

We reckon on the architecture described as a first experience to take Mark Weiser's ideas to the classroom environment with practical technology. Therefore, we are taking these ideas from conception into life, from prediction into reality. The model supposes a challenge as well as an attractive framework to research. We try to extract the implications of ubiquitous computing in computational systems for group learning in particular and group work in general. The ubiquitous computing paradigm has made it possible to adapt the activity of doing exercises in class as a collaborative task. At first sight, it is not necessary to carry out extraordinary changes in the class management. The students still take notes and write in something similar to a small notebook, they frequently direct their attention to the whiteboard, where the teacher displays certain information. There is no need to change position in the classroom and the individual participation remains. But in all this activity underlies the contribution of a series of electronic devices that make this technique possible: collaborative work based on ubiquitous computing.

This learning methodology makes advantageous use of the continuous communication in group and the teacher's individualised attention to each student. Furthermore, at any moment there are access facilities to aids and guidance, in our case study by means of dictionaries, grammar guides and texts available in the system. All this entails a faster assimilation of the group work technique where each student's individual contribution remains effective (and traceable) in the final result. The teacher is still present like in the traditional classroom and, what is more, their participation giving instructions to the whole class as well as commenting and guiding each student's individual proposals can be followed by the entire group through the electronic whiteboard. The students will also follow the teacher's making changes in the organisation and assigning tasks throughout the process. They can even follow the process of the teacher's corrections and final revision, if this is considered advisable.

There is still much to do. The application of our model to other domains and study cases will entail certain modifications in the traditional concepts of teaching, as well as in the most recent theories of learning which try to implement (without much success) the desktop computer as a classroom tool. In this case, we go beyond teaching focused on the individual results and beyond the direct manipulation of the learning tool.

ACKNOWLEDGEMENTS

The authors want to acknowledge theMinisterio de Educación y Cultura, 1FD97-159 project, and the Comunidad Autonoma de Castilla La Mancha for their economic support to this work..

REFERENCES

1. Chong, S.M.Models of Asynchronous Computer Conferencing for Collaborative Learning in Large College Classes. Electronic Collaborators (Bonk, C.J., & King, K.S., editors). 157-182 (1998)
2. Weiser, M., The computer for the twenty-first century. Scientific American, September 1991, 94-104 (1991)
3. Weiser, M., Ubiquitous computing. http://www.ubiq.com/hypertext/weiser/UbiHome.html
4. Norman, D.A., The Invisible computer, The MIT Press, Cambridge Massachusets, 1998
5. Weiser, M., The future of Ubiquitous Computing on Campus, Comm. ACM, 41-1, January 1998, 41-42 (1998)
6. http://www.ubiq.com/parctab/
7. Ortega, M., Computers in Education: the Near Future, Computers and Education in the 21st Century. Plenary Lectures from the Spanish Congress on Computers in Education (ConieD '99), pp. 3-16. KLUWER ACADEMIC PUBLISHERS. Netherlands (2000)
8. Soloway, E., Grant, W., Tinker, R., Roschelle, J., Mills, M., Resnick, M., Berg, R., Eisenberg, M., Science in the Palms of their Hands, Comm ACM, August 1999, 42-8, 21-26 (1999).
9. Vygotsky, L.S., (1978), "Mind in society: The development of higher psychological processes", Cambridge MA: Harvard University Press.
10. Koschmann, T., (Editor), CSCL Theory and Practice of an emerging paradigm. Lawrence Erlbaum Associates (1996).

11. Doise, V., Mugny, G., The social development of the intelect. Pergamon Ed. (1984).
12. Barwise, J., Perry, J., Situations and attitudes, MIT Press Cambridge. (1983).
13. Suchman, L., Plans and situated actions: The problem of human/machine communication Cambridge University Press. (1987)
14. Lave, J. Cognitition in Practice, Cambridge University Press. (1988)
15. Edelson D., Pea R., Gomez L.M., The Collaboratory Notebook. Communications of the ACM, 39(4), 32 (April 1996).
16. Scardamalia M., Bereiter C., Student Communities for the advancement of Knowledge. Communications of the ACM, 39(4), (April 1996).
17. Guzdial M., Kolodner J., Hmelo C., Narayanan H., Carlson D., Rappin N., Hubscher R., Turns J., Newstetter W., Computer Support for Learning through Complex Problem Solving. Communications of the ACM, 39(4), (April 1996).
18. Linn, M.C. Key to the Information Highway. Communications of the ACM, 39(4), (April 1996)
19. Baker, M.J., Lund, K., Flexibly structuring the interaction in a CSCL environment. In Brna, Pl, Paiva, A & Self, J.A. (Eds.) Proceedings of the European Conference on Artificial Intelligence in Education. Lisbon. Editicoes Colibri. (1996).
20. Miao, Y., Haake, J.M., Supporting Concurrent Design by Integrating Information Sharing and Activity Synchronization. In Proceedings of the 5th ISP International Conference on Concurrent Engineering Research and Applications, pp. 1654-174, Tokyo, Japan, (1998)
21. Barros, B., Verdejo, M.F., DEGREE: Un sistema para la realización y evaluación de experiencias de aprendizaje colaborativo en enseñanza a distancia. Revista Iberoamericana de Inteligencia Artificial (9), pp. 27-37 (1999)
22. Redondo, M.A., Bravo, C., Bravo, J., Ortega, M., Colaboración en entorno de aprendizaje basados en casos reales. Aplicación en ambientes de diseño y simulación, In Libro de Actas de INTERACCIÓN 2.000. I Jornadas de Interacción Persona-Ordenador, pp. 143-153. Granada (España) June 2000.
23. Redondo, M.A., Bravo, C., Bravo, J., Ortega, M., Planificación y Colaboración en entornos de aprendizaje por descubrimiento. Un caso de estudio en Domótica. In Actas del la Conferencia International sobre Educación, Formación y Nuevas Tecnologías ONLINE EDUCA MADRID. June, 2000.
24. Bravo, C., Redondo, M.A., Bravo, J., Ortega, M., "DOMOSIM-COL: A Simulation Collaborative Environment for the Learning of Domotic Design". Inroads - The SIGCSE Bulletin of ACM, vol. 32 (2), pp.65-67 (June 2000)
25. Wessner, M., Pfister, H., Miao, Y., Using Learning Protocols to Structure Computer-Supported Cooperative Learning. In Proceedings of the ED-MEDIA 1999 - World Conference on Educational Multimedia Hypemeedia & Telecommunications, Seattle, Washington, pp. 471-476, Association for the Advancement of Computing in Education (1999)
26. http://www.irda.org
27. Redondo, M.A., Sistema de Adquisición de Datos Inteligente. Proyecto Fin de Carrera Escuela Universitaria de Informática Ciudad Real. Universidad Castilla-La Mancha.(1995)
28. Landay, J.A. & Davis, R.C., Making sharing pervasive: Ubiquitous Computing for shared note taking, IBM Systems Journal, Pervasive Computing, Vol. 38, No. 4, (1999)
29. Baber, C., Haniff, D.J., & Wooley, S.I., Contrasting Paradigms for the development of wearable computers, IBM Systems Journal, Pervasive Computing, Vol. 38, No. 4, (1999)

Creation of a Multimedia System for Learning about Oscillations

G. Pedrós, A.Pontes, P. Martínez-Jimenez and A.Amo
Department of Physics, Escuela Politécnica Superior,
Menéndez Pidal s/n, 14004 Córdoba, Spain

Key words: simulation, multimedia, oscillations

Abstract: This paper presents a computer simulation model of a costly laboratory practice, i.e., a vibration bench for making quasi free and damped oscillations. In short, an interactive software has been created incorporating the most representative experiments in this practical exercise. The laboratory instrument simulation is achieved by the visualisation of video images, both of the practical components (loads to be fixed, oils, springs, etc.), and of the experimental process itself. Some random variables have been included in the vibration simulation to reproduce the variability of the experiment conditions. The graphics obtained with the simulation comprise a fictitious velocity of the recorder, permitting students to determine periods and frequencies. After its educational application, we were able to deduce that this new computer tool was fairly useful from a didactic viewpoint.

1. INTRODUCTION

In recent years there has been a wide spread of experimentation with computer-assisted didactic models in the training of highly qualified experts such as aircraft pilots, astronauts and engineers specialised in controls. The same has happened in teaching centres using models which allow the computer to emulate the working of different laboratory equipment and to instruct students how to use it.

The carrying out of these experiments in the Physics teaching laboratory is very costly and requires students to devote considerable time to the study of each of the experiment facilities and to the handling of the equipment.

Thus, some programmes for the simulation of certain important Physics experiments have been created [1], whose worth lies in contributing to the students' capacity to perform, in a simplified manner, those mental actions which are similar to those that they might have to do in a traditional laboratory, with the aid of these models.

In this direction, i.e. the reproduction with software of a costly practice laboratory, we started with an expensive (about 12000 euros) Sanderson vibration bench, used for studying systems which have practically free and also damped oscillations. Traditionally, this experiment simulator is shown to groups of 30 students as a lecture room practice (Figure 1). By using computers, we aim to reproduce this experiment in such a way that students can do it individually and with no cost involved.

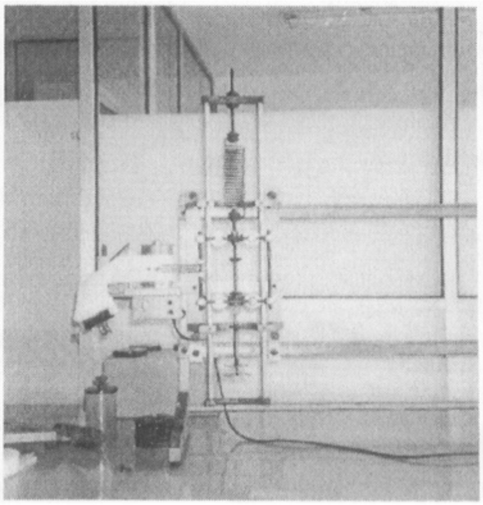


Figure 1. Photograph of the vibration bench

2. SOFTWARE BASICS AND STRUCTURE

When we started to develop this project, two difficulties arose: first, we found very little didactic investigation work related with vibrations even though this theme is potentially of great interest in the teaching of Physics; secondly, there were no educational or simulation programmes covering free and damped oscillations. Those were the reasons that led us to develop this project.

As mentioned previously, the study of oscillations is a useful topic in the teaching of Physics. In pre-university science education special emphasis is placed on the study of uniform and uniformly varied motions and these themes are again returned to in the first years at university, but there is no really profound study of vibrations which, at present, are omnipresent and of great importance in our daily lives. Similarly, in the area of Mechanics, we have noticed that there is numerous research work on students' learning problems and that software have been developed relating to other motions such as the aforementioned uniform and uniformly varied [2], projectile motion [3, 1, 4] or the study of the trajectories of a free throw in basketball [5]. However, there are no works on vibration.

On many occasions, the typical evolution of Physics programmes at university is an inducement to generating almost hermetic attitudes in the fixing of their contents, often relegating to an inferior plane fundamental principles and integrative concepts which are repeatedly applied. However, there is no explanation for the fact that these principles constitute the real pillars on which all our theories are supported and the basis of the mechanisms of reasoning which led to formulate them [6].

We think that vibrations constitute a cohering and transversal theme in Physics since they permit the realisation of an integrated study of the kinetic, dynamic and energetic methods [7], applied to the different types of motions. What is more, oscillations of different magnitudes are common to the study of manifold phenomena in other branches of Physics, such as Acoustics, Physical Optics, Electromagnetism, Atomic and Nuclear Physics, etc. This integrative characteristic of vibrations permits a syllabus design based on the similarities and differences between diverse phenomena by showing how similar models lead to similar equations and to solutions of the same type, and the part played by the characteristics of the physics phenomenon under consideration to justify those very similarities and differences [6].

In some field studies made in the development of Physics, as a subject in the first year of Engineering, it was confirmed that students were confused about basic issues related with vibration, such as: trajectory, position and distance covered; the role of the initial motion parameters; uniform and variable acceleration; energy conservation and dissipation; features of the forces intervening in the motion, etc. For that reason, we consider it necessary for the students to do computer-simulated practicals related with this subject, permitting them to detect these learning difficulties and to overcome them.

The practice laboratory that we have implemented in the computer basically consists of a frame on which springs with different elastic constants can be suspended and loads also added. A piston is placed in a vessel which is filled with liquids of different viscosities and a recorder equipped with a pen traces the different types of motion: quasi-free, underdamped, critically damped and overdamped oscillations. In fact, we have tried to simulate the vibration bench in Figure 1.

In a first approach to a practical treatment of the problem, we used a calculation sheet with graphics to obtain the different types of vibrations simulated [8] but we found out that most students did not achieve the learning objectives predicted. We therefore initiated a teaching project comprising the development, application and evaluation of a tutorial focusing on the study of free and damped oscillations in which we attempted to sum up our opinions on the application of computers in Physics teaching reported in previous papers [9,4].

The essence of this new contribution consists in the design and creation of interactive software incorporating the most important experiments that can be done by students with a free and damped vibration bench. This programme has been created using Visual Basic 5.0 language, with the aim of making the user feel that he/she understands and has a good command of it since we have tried to make it very intuitive and easy to use. Figure 2 illustrates the process of interactivity with the user, by means of input and output values and Figure 3 is a diagram representing the relations with the system.

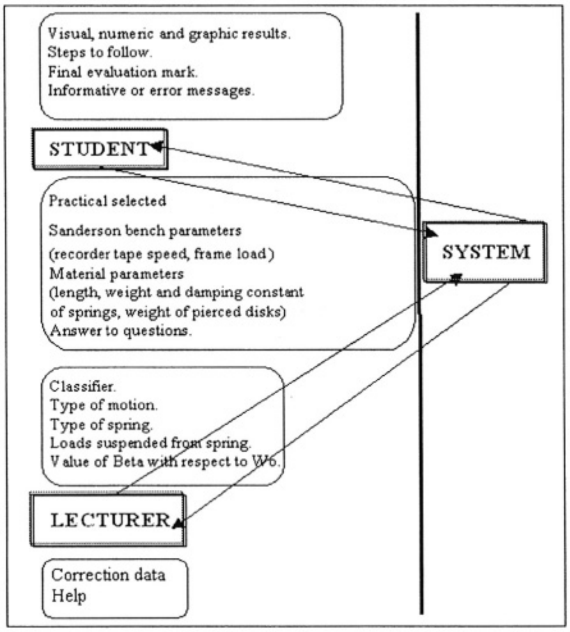

Visual, numeric and graphic results.
Steps to follow.
Final evaluation mark.
Informative or error messages.
STUDENT
Practical selected
Sanderson bench parameters
(recorder tape speed, frame load)
Material parameters
(length, weight and damping constant of springs, weight of pierced disks)
Answer to questions.
SYSTEM
Classifier.
Type of motion.
Type of spring.
Loads suspended from spring.
Value of Beta with respect to W6.
LECTURER
Correction data
Help

Figure 2. Input and output values for the system

The software comprises four parts, which can be accessed from the main menu: Tutorial, Simulation, Introduction and on-line Help.

When the command Introduction is selected, the system shows a screen with four buttons. By pressing the first three buttons in the order established, the vibration bench is shown by means of a series of images and of the explanatory text of the workings of this practical.

The Tutorial module appears on the screen as a hypertext style file showing the theoretical fundamentals of the practical, for reference when necessary. Each page of this Tutorial can also be printed, words sought, etc. "Online Help" is a hypertext file containing help on how to use the programme and the tasks that students have to carry out in the practical.

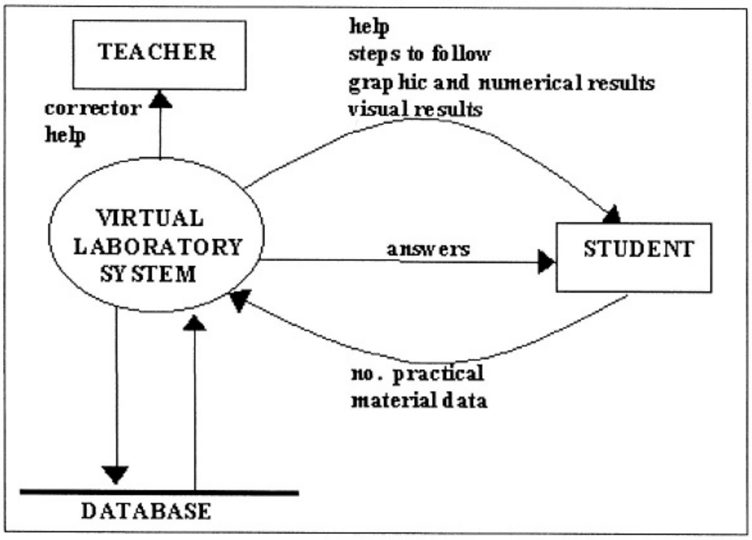


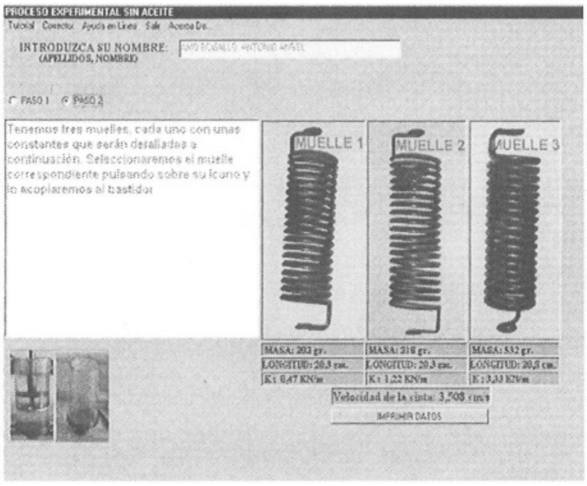
Figure 3. Global flow chart of the programme layout

In the simulation module, a study can be made of the motion of a load suspended vertically from a spring in terms of the following parameters: viscosity, system load and elastic constant of the springs. This module is a virtual representation of the instruments necessary for the student to be able to have it recognised and learned how to use the Sanderson vibration bench when he/she has completed this practical. The simulation is in two parts: 1) experimental process with no oil in which vibrations are obtained with very slight damping and 2) experiment using oil, in which the three types of damped motion are obtained.

In order to create the multimedia system, a video camera film was made. It displayed both the components of the practical (loads to attach, springs, oils, etc.) and the experimental process itself, i.e. how the spring and the loads are fitted to the vibration bench, how the piston is put into the oil, etc.

In this way, a degree of realism is achieved as students can see the practical being done physically through the video images incorporated in the simulation.

For instance, by pressing on Step 2, the student can select the type of spring (figure 4) to attach to the Sanderson bench, and it is also possible to print the data corresponding to the different springs. The spring selection process is done by pressing on the icon corresponding to that spring. Once the spring has been selected, a video corresponding to its attachment process on the vibration bench appears. Steps 3, 4 and 5 also consist of a series of videos, accompanied by an explanatory text on the experimental process which would be done in a Physics laboratory.


PROCESO EXPERIMENTAL SIN ACEITE
Tutorial Conector Ayuda en Línea Salir Acerca De...
PASO 1
PASO 2
MUELLE 1
MUELLE 2
MUELLE 3
IMPRIMIR DATOS

INTRODUZCA SU NOMBRE: (APELLIDOS, NOMBRE)

Tenemos tres muelles, cada uno con unas constantes que serán detalladas a continuación. Seleccionaremos el muelle correspondiente pulsando sobre su icono y lo acoplaremos al bastidor

MASA: 203 gr.	MASA: 216 gr.	MASA: 532 gr.
LONGITUD: 20,3 cm.	LONGITUD: 20,3 cm.	LONGITUD: 20,5 cm.
K: 0,47 KN/m	K: 1,22 KN/m	K: 3,33 KN/m

Velocidad de la cinta: 3,508 cm/s

Figure 4. Spring type selection screen

Next, step 6 consists in the generation of graphic and numerical results. By pressing on this last step of the simulation process, either with or without oil, the data generation screen appears. In Figure 5, as an example, there is a screen showing a motion without oil, since all the screens following the same operating mode and have the same interface. At the top of this screen there are three buttons. The button "Guardar gráfico en disco" (Save graphic in disk) automatically saves the screen in the hard disk, in a BMP format. The directory in which the images are saved bears the name previously entered by the student.

The button "Imprimir" (Print) enables the student to print the register of the corresponding motion. The button "Siguiente" (Next) makes it possible to go on to any new motion resulting from having modified the conditions of the system.

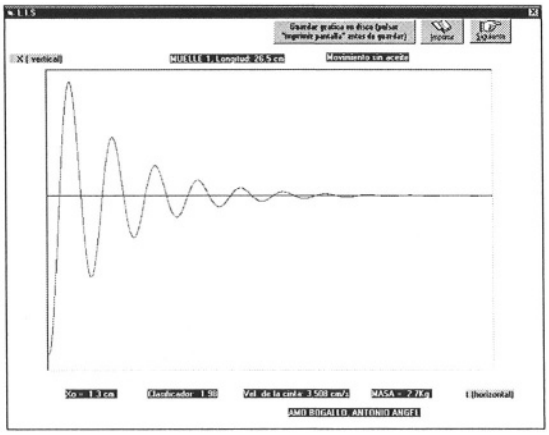

LIS
Guardar grafica en disco (pulsar "imprimir pantalla" antes de guardar)
Imprimir
Siguiente
X (vertical)
MUELLE 1, Longitud: 26.5 cm
Movimiento sin aceite
Xo = 1.3 cm
Clasificador 1.98
Vel. de la cinta 3.508 cm/s
MASA = 2.7Kg
t (horizontal)
AMO BOGALLO, ANTONIO ANGEL

Figure 5. Screen for damped motion generation

In each graphic results screen there appears a series of data characterising the state of the system, such as the length of the spring, the type of motion (with or without oil), the oscillator load, the speed of the recorder tape, etc.

3. EDUCATIONAL APPLICATIONS

We have currently begun to use this virtual laboratory as a supplementary tool to traditional teaching methods, with a view to obtaining more personalised teaching to counteract the present overcrowding of university classrooms. The following are the most important learning achievements for students using this software.

First, the programme permits students to understand the influence of the viscosity, of the system load and of the spring's elastic constant on the different vibrations presented by the system. The graphics obtained by the simulation contain a fictitious recorder speed, letting the student determine the periods and frequencies intuitively. The logarithmic decreases in the amplitude are also obtained by direct measurement on the registers.

It has also been possible to achieve a reproduction of the variability of the experimental conditions by entering random variables which slightly modify the results in the virtual experiment, this variability making it more difficult for the results of two students to coincide. Because of this possible disparity of results and in order that teachers can certify the authenticity of the student's calculations, the system contains a module called "Corrector" only accessible to them. The classification value of the motion to be marked is entered in this module and the programme shows the correct results.

Further, and with the aim of increasing the didactic effectiveness of this tool, a guide has been drawn up, showing how to use the programme when the student is introduced to it for the first time, as well as the many activities to be done as homework with the results obtained: the determination of the experiment constants of the springs, the coefficients of damping, experimental errors associated with the identification of the most relevant parameters in each motion, etc.

As a conclusion, we would like to believe that the new computer tool described in this paper is of greater teaching use than the lecture room experiment on which it is based. However, this hypothesis will have to be confirmed through more educational experimentation and evaluation of the software that we are currently preparing, whose final results will be presented in a future paper.

REFERENCES

1. Valdés, R., Valdés, P. and Pedroso, F. (1997): La realización de experimentos con modelos matemáticos en el ordenador, *Taller Iberoamericano de Enseñanza de la Física Universitaria,* vol.1, pp. 386-393.
2. García, J.J., Pro, A. and Saura, O. (1995): Planificación de una unidad didáctica: el estudio del movimiento, *Enseñanza de las Ciencias,* vol. 13(2), 211-226.
3. Fonseca, M. and Hurtado, A. (1997): Simulación, movimiento de proyectiles con fricción proporcional al cuadrado de la rapidez, *Taller Iberoamericano de Enseñanza de la Física Universitaria,* vol.1, pp. 487-494.
4. Martínez-Jimenez, P., Pontes, A. and Pedrós, A. (2000): Educational Applications of a simulation and solving problems program in the Dynamic area, *Physics Teacher Education beyond 2000*, Barcelona.
5. Martínez, J.A. (2000): Un problema planteado como actividad de investigación: estudio de las posibles trayectorias para el lanzamiento efectivo de un tiro libre de baloncesto, *Enseñanza de las Ciencias,* vol. 18(1), pp. 131-140.
6. Mocoroa, A.B., Zerbino, L.M., Baade, N.N. and Lavagna, M.N. (1997): Propuesta curricular para la enseñanza de la Física en las carreras de Ingeniería, *Taller Iberoamericano de Enseñanza de la Física Universitaria,* vol.2, pp. 178-185.
7. Cárdenas, M. and Ragout de Lozano, S. (1997): ¿Por qué mantener la separation tradicional cinemática-dinámica al ensenar?, *Enseñanza de las Ciencias,* vol. Extra, pp. 383-384.
8. Pedrós, G., Martínez-Jiménez, P. and González-Caballero, D. (1997): El ordenador como instrumento alternative al simulador experimental del estudio de vibraciones en una dimensión, *Adas de la XXVI Reunión Bienal de la R. S. E. F.*, editado por Servicio de Publicaciones de la Universidad de las Palmas, pp. 43-44.
9. Pontes, A. (1999): Utilización del ordenador en la enseñanza de las Ciencias, *Alambique* , 19, pp. 53-64.

Shared Whiteboard Manager and Student Notebook for the PLAN-G Telematic Platform

C. I. Peña de Carrillo[1], R. Fabregat Gesa, A. Urra i Fàbregas, M. Vallès Rocha, J. L. Marzo Lázaro
Broadband Communications and Distributed Systems Group (BCDS)
Instituto de Informática y Aplicaciones Universitat de Girona
Escuela Politécnica Superior, Edificio P1I, Campus de Montilivi, 17071 Girona, España
e-mail: {clarenes, ramon,aurra, mvalles, marzo}@eia.udg.es

Key words: PLAN-G, Adaptive educative hypermedia, virtual teaching-learning, shared whiteboards, notebook

Abstract: The shared whiteboards allow the establishment of a communication-interaction feature between teachers and students to clarify specific terms of the didactic material exposed by means of a web virtual course. Bookmarks and notebooks are navigation tools with historical characteristics in adaptive and educative hypermedia systems [1] that allow the student to take notes on the learning material exposed and to identify and extract relevant references about the web pages visited in order to facilitate their later revision. This work presents the performance of a shared whiteboard manager created to support virtual teaching through the Internet and a student notebook manager that allows students to take private notes on different sections of a web course and to generate bookmark files to export and maintain them as bookmarks through any Internet browser. The virtual environment of teaching-learning used for these tools is the one offered by the PLAN-G[2] project.

[1] PhD student from the Universidad Industrial de Santander (Colombia)
[2] This work was partially funded by the CICYT TEL98-0408-C02-01 and CICYT TEL98-0976 projects. Study and implementation of a ***PLA****taforma telemática de* ***N****ueva* ***G****eneración* to support open and distance learning (PLAN G).

1. INTRODUCTION

The new information and telecommunication technologies, mainly audiovisual, computer applications, telematic technologies and multimedia, allow us to reconsider and reinforce distance teaching and learning systems. *Unidades de Soporte a la Docencia USD* is a teaching and learning system developed by the University of Girona using web technology [2]. USD offers a group of tools allowing teachers to create and publish educational materials, transfer, organise and manage the files of those materials, generate and manage different types of interactive exercises, create and manage teaching units and also to allow students to browse the learning contents and carry out activities using efficient and attractive browsing tools [3].

The shared whiteboard manager [4] was developed for operation on the didactic material generated for USDs with the JVS tool (Java Visual Sequence) [5]. It offers an environment for the creation and publication of interactive presentations made by means of still sequences or related screens that are visualised through a web browser supporting Java language.

The use of shared whiteboards includes another element for communication between teachers and students to carry out interactive and collaborative activities while a teaching and learning material is revised [6]. These activities allow the students to request further explanations on aspects presented in the material. They also allow teachers to carry out the corresponding note or signalling on the same material in order to answer the students' requests. In such case, the system improves the acquisition of on-line knowledge [7].

Taking notes on particular sections of the learning contents offered by a teaching unit is a fundamental activity of distance educational systems and it makes virtual education systems that use web technology more attractive. With the incorporation of the student notebook [8] in the browsing tool for USD, we wanted to go a step further in system adaptability, offering the student a comfortable and private atmosphere to take personalised notes while he/she learns, since, when studying, the revision of notes and the student's own comments is of great help in better understanding the learned concepts.

This work describes the operational structure of the shared whiteboard manager and the student notebook developed to give new functionalities to USD. The architecture and performance of the shared whiteboard manager and for the notebook tool are shown in sections 2 and 3, respectively. Finally, in section 4, conclusions and some proposals for future developments are introduced.

2. SHARED WHITEBOARD MANAGER

The shared whiteboard manager is a guided environment for the visualisation and control of slide shows developed in Java language. In this application, the board is used as a tool for explanation or extension of on-line didactic material included in a USD teaching unit. Using the USD chat application [9], students and teachers that share the same board can communicate. Figure 1 shows the chat request through the board environment. Students are able to connect to a board if a teacher (the guide) is already connected to it.

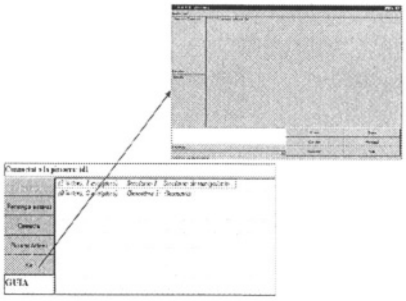


Figure 1. Chat access from the board console

This implementation is a means of teaching support, it guides the teacher in reinforcing explanations; therefore, all the actions (signalling) on slide shows are stored in a database in a selective way for downloading rather than repeating the same explanations and comments in the future. In any case, it is the teacher who decides when to show the explanations made in previous sessions on the current session.

2.1 General Characteristics

The application recognises three types of users:

- The dynamic user, guide or teacher who has the control and the right to carry out any predefined default action on his/her slide show. It also permits assignation of part of the control to connected users.
- The semi-static user or partially guided user who receives user guide permission to control the slide shows presentation on the board.
- The static user or regular student who is only able to visualise the slide show and explanations given by a user guide or a semi-static user.

Depending on the user type, the actions carried out on the board are the following:

– Drawing signs (lines, marks by hand raised, rectangles, etc.) for enhancing, highlighting or pointing out slide information (dynamic user and semi-static user).
– Using the chat application for communication between students and teachers who connect to a board (all users).
– Visualising users connected to the board (dynamic user).
– Opening a slide show and selecting automatic opening for the group of students (dynamic user).
– Transferring control for the semi-static users (dynamic user).
– Saving the presentation with notes and the time used for each generated sequence (dynamic user).
– Recovering notes with or without the stored timing (dynamic user).
– Closing the presentation and, automatically, the one on the shared whiteboard (dynamic user).

User access control to the board is carried out using the USD relational database, consequently saving information regarding visited slides and time spent by the student during each visit.

The performance structure of the shared whiteboard manager is shown in Figure 2.

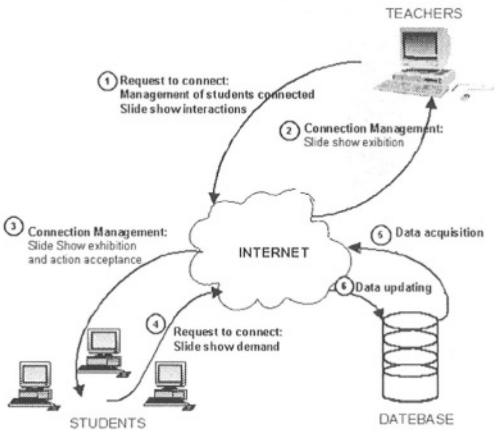

TEACHERS
1 Request to connect: Management of students connected Slide show interactions
2 Connection Management: Slide show exibition
3 Connection Management: Slide Show exhibition and action acceptance
INTERNET
5 Data acquisition
6 Data updating
4 Request to connect: Slide show demand
STUDENTS
DATEBASE

Figure 2. Shared whiteboard manager performance structure

The different procedural phases assigned to each type of shared whiteboard user can be seen in Figure 3.

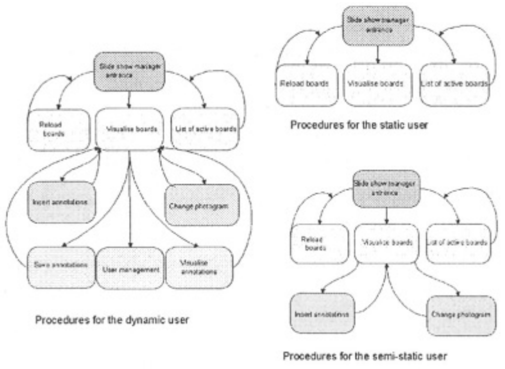


Figure 3. Procedures for each type of shared whiteboard user

2.2 Usage example

A USD teaching unit created with the JVS tool will have an additional tool for communication and collaborative learning; the shared whiteboard manager. Access is through an icon incorporated in the navigational tool bar of the USD environment as shown in Figure 4.

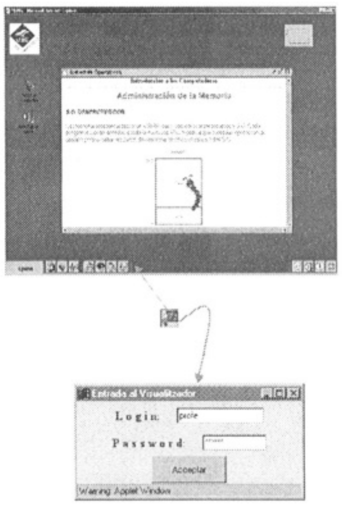
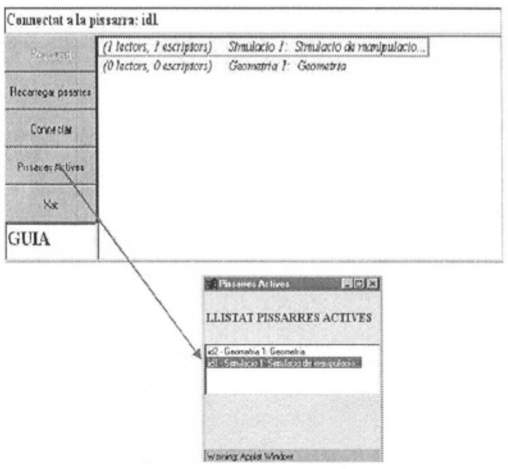


Figure 4. Access environment to the shared whiteboard manager

In order to control access to the application, the system will request a login name and a password that have to match a USD user database.

Once the user has been authenticated, and if the data is correct, he/she will be able to access the shared whiteboard console as it is shown in Figure 5. If a connected guide to the current shared whiteboard already exists, the user will be able to click on the "Pissarres Actives" button in order to obtain the list of available active boards.



(1 lectors, 1 escriptors) Simulacio 1: Simulacio de manipulacio...
(0 lectors, 0 escriptors) Geometria 1: Geometria

id2 - Geometria 1: Geometria
id1 - Simulacio 1: Simulacio de manipulacio...

Figure 5. Shared whiteboard manager console

A shared whiteboard can be opened directly or remotely; that is to say, if a connected guide exists, the student can access the shared whiteboard from his/her local machine or the teacher can open the shared whiteboard for his/her students automatically.

Static users, on the other hand, are not able to act directly on the shared whiteboard. They can only follow the slide show and the teacher notes. An example of a shared whiteboard for static users is presented in Figure 6, where the performance of a computer program is explained. It should be noted that in this case the shared whiteboard toolbar is inactive, as well as the button for the next slide.

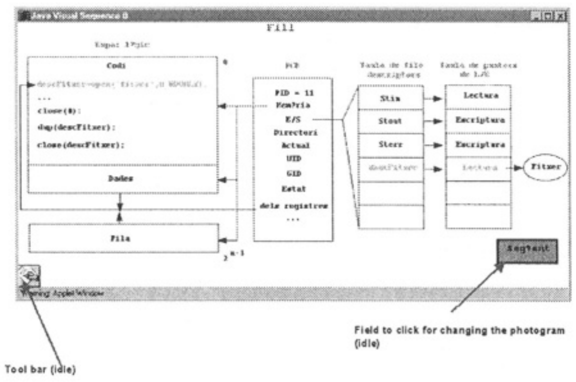


Figure 6. Shared whiteboard for a static user

A semi-static user is able to insert notes on the current presentation and also to access to the next slide by clicking on the "següent" button, if the toolbar appears deployed. The four buttons (left to right) mean: to point something out by raised hand, to use lines or squares to emphasise explanation areas or to insert explanatory text. In Figure 7, an example is given:

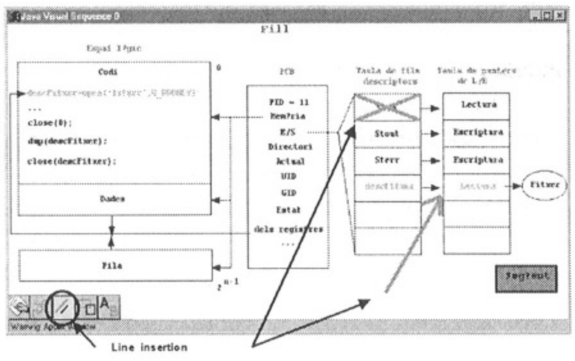


Figure 7. Shared whiteboard for a semi-static user who has inserted lines to highlight some aspects of the presentation

Guide or dynamic users have a broader working environment than static users on the shared whiteboards. Figure 8 shows the additional toolbar available for the dynamic user for carrying out tasks for managing users connected to a shared whiteboard as well as maintenance of the notes in a database.

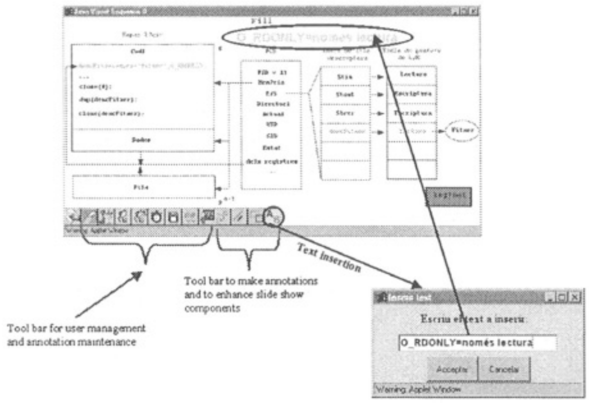

O_RDONLY=només lectura
Text insertion
Tool bar to make annotations and to enhance slide show components
Tool bar for user management and annotation maintenance
Escriu el text a inserir:
O_RDONLY=només lectura
Acceptar
Cancelar
Warning: Applet Window

Figure 8. Working environment for the user guide on a shared whiteboard

By means of the first toolbar created for user management and note maintenance, the dynamic user is able to:

- Delete the current notes made on the slide show exhibited on the shared whiteboard
- Change the border colour of the highlighted area
- Visualise the existing notes of the slide show presentation (current or past)
- Visualise the notes in the sequence they were made
- Initialise timing to recount the sequence of notes
- Save the notes in the database
- Connect the shared whiteboard to particular students automatically
- Assign the control of the shared whiteboard to a semi-static user

With the options on the second toolbar, the dynamic user is able to insert notes and signs as required in order to enhance explanations.

3. STUDENT NOTEBOOK AND BOOKMARKS MANAGEMENT

The notebook allows USD users to take short notes about the web page contents that constitute a teaching unit and to create a bookmark list for each visited page in order to make and export a typical bookmark file for Netscape navigator.

By incorporating the notebook in the USD navigation environment the range of navigational tools is extended. The notebook facilitates the learning and the memorisation process of the concepts from a teaching unit. It has been observed during traditional classes that students get a better grasp of the subject and acquire knowledge more quickly if, in the learning environment, they can summarise or extract the ideas in their own words.

The notebook allows the students to create their own notes when they carry out learning activities inside USD. Right now, these notes are personal and private but, hopefully, later on the possibility to share the notes with other students and to send them by mail to any electronic destination will exist.

The notes are stored in a USD database according to the page where they were inserted and respecting the privacy of the user who made them.

This application has been developed in dynamic HTML using JavaScript, Style Sheets and taking advantage of the DYNAPI libraries [10], which permits working with layers in a dynamic and user-friendly manner.

3.1 General characteristics

The notebook, besides allowing the creation and destruction of personalised notes, offers special tools for:

- Changing the background colour of the note
- Changing the size, colour and style of the font used
- Clearing the note contents
- Inserting links and images into the notes
- Hiding and saving notes
- Listing, deleting and exporting bookmarks

Access to this application is through the navigational toolbar available in a USD teaching unit (See Figure 9).

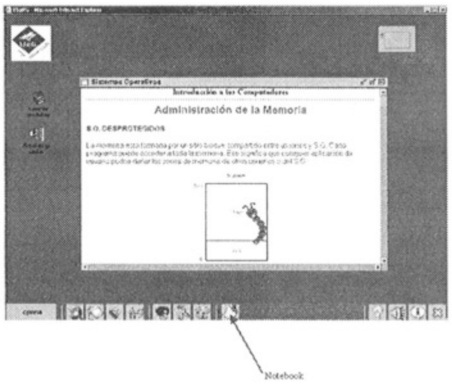


Figure 9. Notebook icon in the USD toolbar environment

When opening a page, a user is able to visualise the saved notes that he/she has taken in previous sessions (all or some) as well as to create new ones, linking them or modifying them.

The application offers the possibility of defining the properties of the notes, including size, background colour and size and colour of the font. Default properties are: yellow background, font size 14, and font colour black.

Besides inserting text into a note, the application also allows the insertion of images and links, not only on the current page but on external pages of the USD teaching unit as well. If the user wishes so, the external links can be stored in an address book, available at any time if it was previously created in the same teaching unit and in the current page. The address book will create a typical bookmark file, easily exportable to Netscape navigator.

3.2 Usage example

To begin taking notes, to create notes and bookmarks maintenance, or to visualise them, the user clicks on the notebook icon of the USD toolbar from its virtual desktop (see Figure 9). Figure 10 shows the set of tools available in this application.

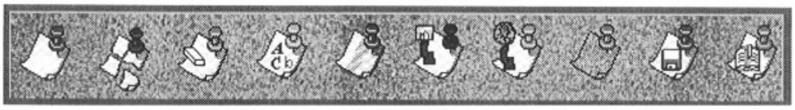


Figure 10. Student Notebook Toolbar

The icon features are described below:

This is the only icon that the user can use if the user has not created any note at all. The default size for new notes is 200x200 pixels.

To delete a note from the database.

To erase the contents of the current note. In this case, the note properties will be the same as the default ones.

To change the background colour of the note.

To choose and change the font properties of the current note (size, colour and style font).

To insert an image into the note.

To insert a link into the note.

To hide the current note.

To store the note in the database.

To manage the bookmark list.

Figure 11 shows the features of some of these icons.

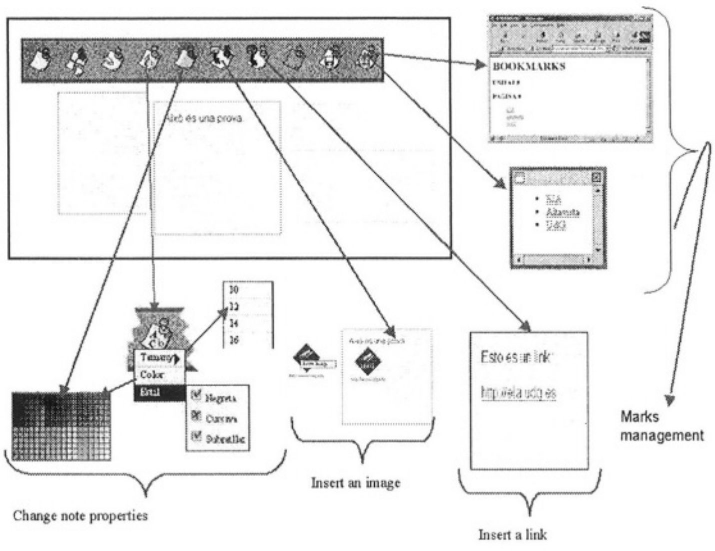


Figure 11. Student notebook toolbar showing some icon operations

4. CONCLUSIONS AND FUTURE WORK

The shared whiteboards system is intended as a support for collaborative work and communication between students and teachers in carrying out an on-line task with USD learning materials.

With the notebook application, the student will have complete freedom to make summaries of and comments on the learning material offered in his/her own words, a fundamental element in improving memorisation and acquiring knowledge.

The development of these new functions for the USD platform implies an improvement in the student's working environment for learning activities of a USD teaching unit.

Currently, the shared whiteboard manager works independently of the USD environment. We are now testing it in a local area broadband communication network and distributed systems group of the University of Girona. The next phase consists of incorporating this tool within the communication module of the USD environment. It will also allow students to choose for visualising the explanations made on a shared whiteboard in past sessions and to include simultaneous user interaction on the same shared whiteboard, requiring a high degree of synchronisation of activities to achieve the maximum efficiency.

As for future extensions of the notebook application, we plan to integrate it into the USD information search engine for selective searches of notes or bookmarks using keywords and to allow sharing the notes between students, subject to the owner's permission.

REFERENCES

1. Brusilovsky P. (1996). "Methods and Techniques of Adaptive Hypermedia"; User Modeling and User-Adapted Interaction, 6, 2-3 (1996) 87-129.
2. Fabregat Gesa, R., Marzo Lázaro, J.L., Peña de Carrillo C.I. (1999). PLAN-G: Plataforma Telemática para la Enseñanza Abierta y a Distancia Utilizando el Web como Soporte, Revista de Enseñanza y Tecnología Nro. 15, Asociación para el Desarrollo de la Informática Educativa, ISSN 1138-7386, pp. 27-41, España, 1999.
3. Peña de Carrillo C.I., Fabregat Gesa R., Marzo Lázaro J.L. (2000). WWW-based Tools to Manage Teaching Units in the PLAN-G Distance Learning Platform, EDMEDIA 2000, World Conference on Educational Multimedia, Hypermedia & Telecommunications, Association for the Advancement of Computing in Education, ISBN-1-88-00-94-40-1, Montreal, July 2000.
4. Urra i Fàbregas A. (2000). Aplicació client-servidor per a la gestió de pissarres compartides. Treball Final de Carrera – Enginyeria Tècnica en Informàtica de Gestió E.T.I.G., Universitat de Girona, Escola Politècnica Superior, Girona.

5. Portugal Brugada A., Fabregat Gesa R., Marzo Lázaro J.L. (1999). "Java Visual Sequence: Generador de Presentaciones para Internet". Proceedings CONIED'99. Congreso Nacional de Informática Educativa. 17-19 de Noviembre de 1999. Puertollano. Ciudad Real. España.
6. Nicol, J., Gutfreund, T., Paschetto, I, Rush, K. & Martin, C. (1999). How the Internet Helps Build Collaborative Multimedia Applications, Communications of the ACM, Volume 42, Number 1, pp. 79-85, January, 1999.
7. Hmelo, C., Guzdial, M. & Turns, J. (1998). Computer-Support for Collaborative Learning: Learning to Support Student Engagement, AACE Journal of Interactive Learning Research (JILR), Volume 9, Number 2, p.107, 1998.
8. Vallès Rocha M. (2000). Gestió de notes i bookmarks per al PLAN-G. Treball Final de Carrera – Enginyeria Tècnica en Informàtica de Gestió E.T.I.G., Universitat de Girona, Escola Politècnica Superior, Girona.
9. Pancorbo i Creus L. (1999). Xat: Mòdul de Comunicació Síncrona, amb suport per a diferents idiomes, sola plataforma web. Treball Final de Carrera - Inginyeria Tècnica en Informàtica de Gestió E.T.I.G., Universitat de Girona, Escola Politècnica Superior, Girona.
10. Steinman D. (1996). http://www.dansteinman.com/Dynapi

Guiding the User

An element to aid knowledge construction in adaptive hypermedia systems

Tomás A. Pérez, J. Gutiérrez, R. López, A. González & J. A. Vadillo
University of the Basque Country (UPV-EHU), Dept. of Languages and Computer Systems, Computer Science Faculty, P. O. Box 649, 20080 San Sebastián, Gipuzkoa.

Key words: Hypermedia, adaptation, constructivism, adaptive hypermedia systems, interactive learning hyperenvironments

Abstract: When developing intelligent tutoring systems, we design plan modules to establish which order the concepts are to be presented to the student. This option is feasible because these systems retain the control over navigation through the sea of contents. In hypermedia systems the student has the control. They decide which contents they want to visit next. In this paper we present a way of adapting those didactic planning modules of intelligent tutoring systems into help elements for those students with problems in adaptive hypermedia systems for education. Our main objective is to preserve constructive characteristics of the system.

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) are generally based on instructive theories: they have a series of predetermined didactic objectives to cover and they plan the strategies so that the student can achieve them. A plan shows the order in which the contents are presented and the messages given. ITS also evaluate the feedback that the student provides and they update the student's data from it, and if they consider it necessary they re-plan to achieve the outlined objectives [1].

Interactive learning hyperenvironments (ILHE) are closer to constructive learning theories [2]: students have control over the contents they want to visit, they establish the visiting order and define their learning objectives.

The system can probably establish certain methods to "motivate" learning, such as activities [3].

The decision to use hypermedia technology implies yielding control over the student's navigation through hyperspace. This is one of the points confronting instructive and constructive theories defenders. Marchionini [4] indicates that an undisciplined navigation through the contents can disorient and even confuse the student. It is true that surfing through hyperspace requires responsibility and the capacity to make decisions, characteristics that any defender of the constructive paradigm considers necessary to produce learning.

However, there are not only advantages with these systems. The use of hypermedia, systems has certain risks that the content designer and users should assume: hypermedia illiteracy, problems in creating hyperspace, cognitive overload, and issues on orientation and getting lost in hyperspace.

Hypermedia systems impose a special way of working that has not yet been standardised and which differs from traditional learning methods. For this reason, all users require a certain familiarisation period which is usually called *hypermedia literacy* [5]. Among other things, they should get to know the user's interface and more specifically, the *look & feel* of the application in which they are working. Fortunately, this problem is being overcome by the appearance of browsers over HTML pages via Internet. Although some experts in hypermedia defend the fact that many of the possibilities considered by experts are not admitted, like those which are reflected in the DEXTER hypermedia systems reference model [6].

The low familiarisation with this type of systems makes the creation of hyperspace more complex for authors [7]. There are many elements that must be considered, now that the documents do not have a lineal reading sequence. On the one hand, context-independent nodes should be offered so that the user cannot feel de-contextualised when accessing them because the author has assumed that a way has been followed different from the expected one. It is important to anticipate this type of problem and, even more important, to avoid the tendency to use sequencing nodes to eliminate that situation.

The problem of *cognitive overload* [5] may happen if too many links have to be considered. This problem has two sides depending on whether we consider the author or the user of the hyperspace environment. The first one, as previously mentioned, occurs because the author must consider all possible paths that could lead to a particular node and at a particular moment they are not able to control them all. From the point of view of the user of that hyperspace, they can decide to take certain "detours" during the search for a particular content. Of course, the user has to remember to return the appropriate way. If the detours are nested ones, one inside each other, there will be a moment when the number of links to be handled would be very large.

In hypermedia learning systems, as the student decides (or is able to decide) which contents to see next, they need to know where to find these contents. This means that it is necessary for the student to create a mental map with the organisation of the domain that is presented in order to be able to locate the information that interests them within the hypermedia with the minimum effort [8]. This map will help them to be guided through the contents and to carry out the learning just as they wish. Complexity of domains at certain times can cause the student to wander through hyperspace without actually knowing exactly where they are or how to reach the part of the domain contained in their mental maps. This problem is known as being *lost in hyperspace* [9].

It is therefore necessary to help students make sure that they reach the nodes they have interest in and that they find elements in hyperspace that help them to make the decisions to successfully travel through hyperspace.

Not all students need the same help. Some need more than others [10]. Therefore, individualised attention is interesting as regards the assistance given, so that the path taken is effective. Based on the conclusions of several studies analysed, Tergan, [11] states that the amount of guidance given to the student should be something that should not be completely decided by the student since they do not usually choose the most appropriate guidance for their characteristics. In particular, Ross & Morrison [12] show that, when they are given the opportunity to choose, students that need more help are generally those who choose lower levels and vice versa, those who do not need it choose higher levels.

In this paper we present a simple way of offering guidance based on the structure of the hyperspace and an organisation of the nodes when it is created. Even though the guidance system can work independently, it is designed to be included inside a learning ILHE like Hezinet [3], it considers more elements in the student's model than are presented here, some of them are also presented in this book [13]. In the following sections we present the designed guidance module, starting with the concepts on which it is based and showing an example that illustrates its operation inside an example hyperspace.

2. GUIDING THE USER

To produce the adaptation of the navigation assistance, we have defined guidance levels that establish the basis for the recommendation of links to be followed by the student. However, the students will always be able to decide if they want to follow the suggestions given or not. The guidance offered does not always have the same intensity, at certain times it is reduced; depending on the itinerary the user is following, as we explain later.

A *guidance level* consists of the organisation of the nodes in the hyperspace into groups with certain coherence. These groups, which we call *hyperdocuments*, are associated to each other using pedagogic relationships in the same way as the pedagogic domain of an intelligent tutoring system does. The system will use these relationships to establish suggestions to the student.

We will consider that a hyperdocument (from the guidance point of view) is a group of nodes among which it is not necessary to recommend links to follow. The system will only make suggestions over links that go from one hyperdocument to another. The greater the hyperdocuments, the smaller the need to guide the student. In practice, depending on the number of nodes in a hyperdocument, it can include nodes on one concept, one point of view on a subject, all the points of view on a subject, etc.

To illustrate the idea of guidance, the following figures show three different levels of guidance (identified as Gl, G2 and G3) for a particular portion of hyperspace or a hyperspace environment. This hyperspace is formed by the nodes and links between them. The circles in the figure, with names represented by letters from A to H, correspond to the nodes, and the arrows between the nodes identify the links. To avoid naming the links, we will use the name of the origin node and the destination node of each one of the arrows, to refer to them. In this way, the link $A \rightarrow B$ is the link represented by the arrow whose origin is node A and destination is node B. Likewise, we will abbreviate the representation of the links that have a common destination by placing all the destinations after the corresponding arrow. For example, $B \rightarrow CDH$ represents the links $B \rightarrow C$, $B \rightarrow D$ and $B \rightarrow H$. In the same way, we will abbreviate the links with a common destination. Therefore $BC \rightarrow D$ refers to the links $B \rightarrow D$ and $C \rightarrow D$. Finally we will represent the hyperdocuments with shaded rectangles. To refer to the hyperdocuments we will make reference to the nodes that the hyperdocument contains, so that the hyperdocument AB is the one that contains the nodes A and B.

The following figure represents guidance level Gl (left). It is an extreme level in which a hyperdocument has been defined for each one of the nodes in the hyperspace. On level G2 (on the right) wider hyperdocuments than on level Gl have been defined as we can observe in the following figure. The nodes and links are the same because the hyperspace has not changed; the only thing that has changed is the way the nodes from the previous case are grouped. In this case four hyperdocuments have been defined. We will identify them by the nodes that compose them: AB, CDF, E and GH.

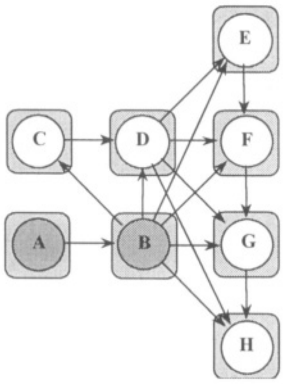


Figure 1 Guidance level G1

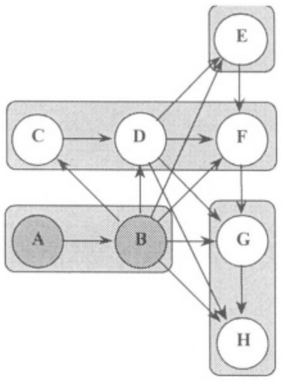


Figure 2. Guidance level G2

Finally, at level G3 there is only one hyperdocument that includes all the nodes in the hyperspace.

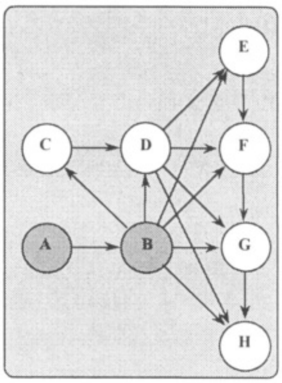


Figure 3. Guidance level G3

To illustrate the operation of the guidance module we will use the hyperspace previously presented with those three guidance levels defined. The guidance system can act in three different ways for each link: (1) recommending the use of that link; (2) dissuading its use; and (3) neutral behaviour in which no type of suggestion is established. The first two are used when there are links between two hyperdocuments (alternately) whilst the third is used when the link is inside the current hyperdocument.

On level **G1** assistance is given by recommending or dissuading the following every link of the node where the user is currently situated (A→B, B→CDEFGH, C→D, D→EFGH, E→F, F→G and G→H) because all the links are between hyperdocuments. The guidance message has more or less help depending on the knowledge about the nodes visited in the hyperdocument as we show later on.

On level **G2** there is no guidance in the links whose origin and destination are in the same hyperdocument. This happens in link A→B inside hyperdocument AB, or link D→F in hyperdocument CDF. Therefore, guidance is only given in links B→CDEFGH, D→EFGH and F→G.

On the other hand, when we are on level G3, the system does not provide any assistance, since there is only one hyperdocument that includes all the links and, therefore, there are no links between nodes of different hyperdocuments.

The guidance levels will give us hints to advise the student to follow a link (F), dissuaded them (D) or not make any type of recommendation (N). The following table shows a summary of the assistance in the form of guidance the system would provide. To simplify the process, let us suppose that the system has analysed the links in the hyperspace, the relationships between the hyperdocuments and the student's particular characteristics and has decided that the visiting order of the recommended hyperdocuments[6] is A-B-C-D-E-F-G-H (alphabetical order) on level Gl, AB-CDF-E-GH on level G2 and ABCDEFGH on level G3. Among the navigation characteristics associated to the student we know that they are in node B and that they have previously visited A. In the following paragraphs we describe the table in more detail.

Table 1. Recommendations on the links from node B

	link B→						
Guidance	**A**	**C**	**D**	**E**	**F**	**G**	**H**
G1	N	F	D	D	D	D	D
G2	N	F	F	D	F	D	D
G3	N	N	N	N	N	N	N

On guidance level **G1** the visit to node C is recommended, which is the node that contains hyperdocument C, which follows hyperdocument B in the recommended path. All the references to the other nodes (D, E, F, G, and H) are dissuaded except the link to the nodes in hyperdocument A (that is, to node A), which would correspond to an already visited node and therefore the negative recommendation would become neutral (we assume that the student knows the organisation of the hyperdocuments which they have come from).

On level **G2** they are advised to visit nodes C, D and F, all of which belong to the same hyperdocument (CDF), which is the one that follows

[6] To simplify the example, we have considered that there is only one recommended path. The system could decide that there is more than one possibility of visiting at a given time.

hyperdocument AB according to the sequence that has already been set from node B, where the user is found. It also dissuades visiting the nodes in the other hyperdocuments, since they will be recommended later on. As we have previously mentioned, the suggested nodes do not have to belong to only one hyperdocument, there can be several. This decision has been taken to simplify the illustration of the system operation. For example, you could consider that after AB, the advisable hyperdocuments would be CDF and GH. In such a case, the system could advise you to go to one of the nodes in those hyperdocuments.

Finally, at level G3 there is only one hyperdocument and, therefore, no indication will be given in the links offered.

2.1 Intensity of the guidance

The guidance that has been presented until now can offer certain disorientating effects in the surfing. The problem arises from the neutral behaviour with regards to the links with destination in nodes of the actual hyperdocument as opposed to those that end in other hyperdocuments. Table 2 shows a conflicting itinerary carried out in the hyperspace that we are using as an example with an intermediate level of guidance such as G2. We will discuss this matter more thoroughly in the following paragraphs.

Table 2. Example of conflicting sailing for the hyperspace

Step	Current node	F	D	N	Decision
(1)	To	-	-	B	A→B
(2)	B	C, D, F	E, G, H	A	B→D
(3)	D	E	G, H	B, F	...
(3')	D	F^+E^-	G, H	B	

The starting point is node A (Step 1). As we are in guidance level G2 and the link A→B is internal to hyperdocument AB, the system's response is neutral as regards recommendations. Since the only possibility that exists is to follow one link, the decision made is to cross this link and reach node B.

Once at node B there are now more navigation possibilities represented by links B→CDEFGH. Assuming the previously mentioned recommendation order, the system recommends following the links that lead to hyperdocument CDF and it dissuades the links that lead to the other hyperdocuments (E and GH). In this position the user decides to follow the link B→D.

The link D→E is recommended from node D, the links D→GH are dissuaded and the links D→BF remain neutral. The recommendation made at this time is problematic. It is not coherent that after arriving at a hyperdocument with more than one node you intend to visit another hyperdocument without first visiting the contents included in the current hyperdocument. The solution to this problem is given through the creation of an additional behaviour to the guidance that reinforces or reduces its intensity. This means that the guidance that is offered does not always have the same strength, it depends on the itinerary that the user follows. The intensity grows when the current hyperdocument is known.

The *intensity of the guidance* defines the degree of rotundity in the assistance offered by the guidance module. It should be less convincing when they have just arrived at a hyperdocument than when they have been navigating through it for some time.

The alternative behaviour of the system in Step 3 is reflected in Table 2 as (3'). Once they have arrived at D (the first node in the hyperdocument CDF) the recommendation to go to E has to have less intensity (represented by the sign '-') than when they visited nodes C and F. In this way, the nodes in the current hyperdocument (in this case only F, represented by a '+' sign) are placed in front of those in other hyperdocuments that can be reached, in the list of recommendations. In any case, the student can ignore the indications proposed and choose the path that best suits them, since the control is given to them.

The system will be guided to establish the student's knowledge on the contents of a hyperdocument in the nodes visited, the time that they have spent in each one of them. It is also possible to add other parameters if the system analyses the knowledge acquired by the student.

2.2 Initial level of guidance and changes in levels

One of the decisions that will be important when checking the behaviour of the students when using the system is to determine the initial state of the level of guidance that the system provides. If there is no previous data, the error is minimised by offering an intermediary level of guidance, which can be corrected after observing the students' behavioural patterns with the system. If the system is offered for different pedagogic domains, the initial level can be based on the level that they had in the domains that have been previously worked in.

When the guidance was designed the possibility that the guidance levels could change during the course of the navigation through the hyperspace was considered. In our case, a change on the level of guidance implies that the hyperdocuments that must be considered to propose the help also change and therefore the system associated behaviour changes.

When you go from one level with larger hyperdocuments to another level with smaller hyperdocuments the level of assistance offered is greater. This will happen when you go from level G2 to level Gl. There will be a greater number of suggestions offered after the change.

If the change on the level of guidance implies that the new hyperdocuments are larger (for example, from Gl to G2), the orientation offered will be smaller.

In general, the level changes can cause certain disorientation when changes in the student's expected behaviour take place. If help appears that did not previously exist in a link, the student may think that it is a new link. The same thing can happen if the assistance given disappears. Therefore in this module it is a rule to warn the student so that they are aware of the changes that have taken place and they can find an explanation in the changes that take place when working with the system. In any case this point is not considered problematic if the level changes are established because they will always be based on the initiative that the student demonstrates.

3. CONCLUSIONS

In this paper a new method is presented to offer help to the users of a hypermedia system based on the *level of guidance* and *hyperdocument* concept. To aid interaction with the user an order of recommendation for the links is not established (even though it does exist), it helps just by using the suitability of surfing through some of the links that are available to the student.

The system does not impose any restrictions on the student who surfs through the hyperspace, so that the constructive behaviour of the system is preserved. It only offers assistance to those who do not have enough initiative or motivation to accept the responsibility implied by the control offered when using a hypermedia system.

The system is always present, it offers its help even if the student ignores it. Some systems, such as ELM-ART [14] eliminate the suggestions as soon as the student ignores them. In the definition of this help system we have considered the fact that the student can de-activate the help or reactivate it (at the level that has been assigned to them) when he/she feels it is necessary. Nevertheless, help is always offered. The system assumes that if the student does not pay attention to the suggestions it is because their own initiative advises them to follow the links. However, the later help will be affected by the graduation of the intensity in the recommendations.

The system proposed is not only a simple way to work in a hyperspace that is not proven, but it is one of the possible improvements to install in the Hezinet system [15].

REFERENCES

1. Gutiérrez, J. (1994). *INTZA.* Tesis Doctoral. Universidad del País Vasco UPV-EHU. Donostia
2. Gros, B. (1995). Nuevas tecnologías, viejas polémicas: el recorrido interminable por el dilema instruir-construir. *Substratum,* vol. II, No. 6, 95-111.
3. Pérez, T. A. (2000). *Un hyperenvironment adaptativo para el aprendizaje instructivo/constructivo.* Tesis Doctoral. Universidad del País Vasco UPV-EHU. San Sebastián.
4. Marchionini, G. (1990). Evaluating Hypermedia-Based Learning. En David H. Jonassen & Heinz Mandl (Eds.). *Designing Hypermedia for Learning.* NATO ASI Series, Springer Verlag: Berlin.
5. Nielsen, J. (1993). *Hypertext & Hypermedia.* Academic Press: Cambridge, Mass. (USA).
6. Grønbæk, K. & R. Trigg. (1994). Design Issues for a Dexter-based Hypermedia System. *Communications of the ACM,* 37 (2).
7. Garzotto, F., P. Paolini & D. Schwabe. (1993). HDM: A model-based approach to Hypertext application design. *ACM Transactions on Office Information Systems.* Vol. 11(1).
8. Jonassen, D: H. & R. S. Grabinger. (1990). Problems and Issues in Designing Hypertext/Hypermedia for Learning. En Jonassen & Heinz Mandl (Eds.). *Designing Hypermedia for Learning.* NATO ASI Series, Springer Verlag: Berlin.
9. Nielsen, J. (1998). *Usability Engineering.* Academic Press: London.
10. Wager, W & M. D. Gagné (1988). Designing Computer-Aided Instruction. En D. H. Jonassen (Ed.). *Instructional Designs for Microcomputer Courseware* Lawrence Erlbaum Associates Inc. Hillsdale, NJ (USA).
11. Tergan, S. O. (1997). Misleading Theoretical Assumptions in Hypertext/Hypermedia Research. *Journal of Educational Multimedia and Hypermedia, 6* (3/4), 257-283.
12. Ross, S. M. & G. R. Morrison. (1989). In search of a happy medium in instructional technology research: issues ;concerning external validity, media replications and learner control. *Educational Technology Research and Development,* 37 (1), 19-33.
13. Pérez, T. A., J, Gutiérrez, R. López, A. González & J. A. Vadillo. (2001). Pedagogical Strategies with Hypermedia: Limiting access to hyperspace for educational purposes. In M. Ortega & J. Bravo, (eds.) *Computers and Education: Towards an Interconnected Society.* Kluwer Academic Publishers, Dordrecht, Netherlands.
14. Schwarz, E, P. Brusilovsky & G. Weber. (1996). World-Wide Intelligent Textbooks. *Proceedings of the first World Conference on Educational Telecommunications, ED-TELECOM'96,* Boston, MA, AACE: Charlottesville, Virginia (USA), 302-307.
15. López, R., J. Gutiérrez, A. González, J. A. Carro & T. A. Pérez. (1999). Cómo aprender euskara a distancia de forma adaptativa. *Actas del congreso nacional de informática educativa CONIED'99.* M. Ortega & J. Bravo, (eds.), Puertollano, Ciudad Real (Spain).

Pedagogical Strategies With Hypermedia

Limiting access to hyperspace for educational purposes

Tomás A. Pérez, J. Gutiérrez, R. López, A. González & J. A. Vadillo
University of the Basque Country (UPV-EHU), Dept. of Computer Languages and Systems, Computer Science Faculty, P.O. Box 649, 20080 San Sebastián, Gipuzkoa

Key words: Hypermedia, adaptation, constructivism, adaptive hypermedia systems, interactive learning hyperenvironments

Abstract: Creators of hypermedia documents for interactive learning hyperenvironments for instruction frequently need to establish certain restrictions on the navigation in hyperspace. They are normally influenced by a specific learning strategy. For example, they may not want a student to visit nodes on multiplication before visiting nodes related to addition. In traditional systems those restrictions are established by limiting the existence of the elements that make the relationship between concepts effective: links. In this paper we propose a simple way of limiting access, by avoiding the creation of incorrect mental maps of hyperspace by the student.

1. INTRODUCTION

Creators of hypermedia documents for interactive learning hyperenvironments (ILHE) frequently need to establish certain restrictions on the navigation in hyperspace. They are usually influenced by a specific learning strategy. For example, they may not want a student to visit nodes on multiplication before visiting nodes related to addition. In traditional systems those restrictions are established by limiting the existence of elements that make the relationship between concepts effective: *links*. There are at least four methods of carrying out this limitation. They are described as follows and summarised in Table 1.

Table 1. The negative impact of the different methods for limiting the surfing in hyperspace on the student's orientation and cognitive load.

Method	Orientation	Cognitive load	Observations
1	Many links	Many links	Links are not eliminated
2	Links missing	Less links	Incomplete mental models
3	Many links, some non-active	Many links	Cognitive load
4	Links missing	Links appear!	Incorrect mental maps of hyperspace

The first method does not limit navigation. All links are shown in every node. It is the method that defenders of constructive theories would propose [1, 2] since they suggest that it is the student who actually defines the learning strategy. This decision implies an increase in the cognitive load the student has to work with, which can be counterproductive [3], especially when the number of options increases after each link visited. Likewise, if appropriate links are not chosen the existence of so many links may result in disorientation.

The second method consists in eliminating the links contrary to the pedagogical strategy being used. In this case, it is an instructive pedagogical strategy. Within the nodes, it eliminates those links to nodes which, according to the prevailing pedagogical strategy, should not be presented at that moment. This presupposes that the system controls the student's path along hyperspace and only provides suitable links in order to present the concepts appropriately. This option provides the least cognitive load because links that would appear in other cases are eliminated. In addition, the reduction in links means that there are fewer possibilities that new learners of the subjects could become disorientated. In any case, the main result of this method is that the pedagogical strategy may avoid the appearance of links that associate related ideas [4]. This situation can lead the students who partially know the domain to feel disorientated by the non-existence of certain links and the new students to build mental models from the incomplete domain that will lack the relationships that do not appear.

The third method consists in showing all the available links, but deactivating certain anchors, those which meet certain conditions defined a priori such as, for instance, having previously visited the destination node, meeting the conditions of a particular pedagogical strategy, etc. This system is a mixture of the two previous methods, since it shows all the links but only makes some of them available. The interface should be able to differentiate each one of them. The student will then choose a link from the available links shown. As they advance through hyperspace, the conditions will be different and the visit to a particular node can cause the activation of more links in that node. This method reduces the student's cognitive load to levels somewhat higher than method 2, because even though more links are shown here, only some of them are available.

The fourth method consists in hiding certain links. The node has all links but the system only shows those that meet certain conditions defined a priori. Before presenting the node the system checks if each link meets the conditions and it decides to present it or not, as appropriate. As in the previous method, the interaction with the system will make conditions vary and therefore the number of links shown in a particular node over the time will also vary. Cognitive load provided in this case is smaller since there are no unusable links. However, this can cause certain disorientation problems to the student.

The disorientation that may be caused is the same as if a person visited a place they did not know, such as the garden in Figure 1 twice. They can notice details such as the doors, the trees, the walls, the rest of the landscape...

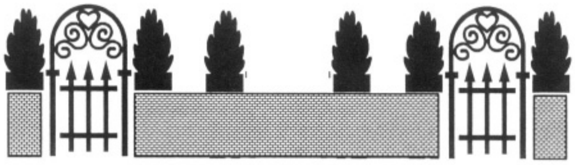


Figure 1. The garden of a house, is it the same as the one in Figure 2?

If the second time they visit the same place, the landscape has changed, because there is an extra door, like in Figure 2, it will be difficult to recognise it as the same place that they had previously been to. However, if details such as which doors were closed and which were open had changed from one visit to the next, the colour of the doors, or the trees had been pruned... they would not get lost so easily.

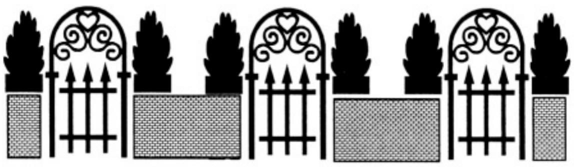


Figure 2. The same garden as in Figure 1, showing the door which was hidden

This same thing happens with nodes and hiding links. If between two visits to one node, links appear or disappear, the students will encounter more difficulties in locating themselves within hyperspace. We must remember that these paths through hyperspace are usually to discover new nodes and one node with more links may be mistaken for a different one, thus causing the mental map of hyperspace to be incorrectly built.

We consider that if it is necessary to establish access limitations in hyperspace of an ILHE, method 3 is the most suitable method. This option is flexible enough to change to method 1 and methods 2 and 4 if it is complemented with a tool controlled by the student (not the system) that enables hiding links which meet certain conditions (those visited, those that are not operative, etc.). In this way it is the actual designer of that portion of hyperspace who decides on the behaviour of the system [5].

The following sections show a method we have devised to provide certain limitations on the access to hyperspace as a result of the adaptation to pedagogical strategies that can be proposed. This method is included in the Hezinet system [6]. A fictitious hyperspace has been used to simplify the presentation of the links, avoiding semantic problems when working with them and locating the necessary examples without making it more complex.

2. ADAPTING A PORTION OF HYPERSPACE TO PEDAGOGICAL STRATEGIES

In order to limit access to certain parts of the hyperspace the author is asked to define the free navigation areas (FNAs). *A free navigation area* is a group of nodes usually related by the concepts that they cover through which the user can surf freely.

The extent of FNAs indicates the amount of control given to the student. It depends on the user's cognitive capabilities, the structure of the actual domain and the dexterity with the surfing interface that the system has [7]. The greater the dexterity with the interface (or the simpler the interface is), the greater the FNAs a given student can control [8].

The group of FNAs covering the whole pedagogical domain (hyperspace) constitutes a level of accessibility. The operation of this module is based on them to establish limitations on the surfing. Different levels of accessibility can be defined for one same hypermedia document, as many as degrees of adaptation to the user in the limitation to be created. These levels of accessibility can coincide with a different organisation of the nodes to be established (lessons, chapters, wider concepts, etc.).

2.1 Operation of the accessibility limitation module

The module that limits the width of the hyperspace is based on the definition of the levels of accessibility mentioned in the previous section.

One particular user always stays on one of those levels. A partial ordering of the navigation between the different FNAs must be given, based on the relationships that have been entered to implement the required pedagogical strategy. This ordering can be directly defined by the author of the hyperspace or deducted by another intelligent module by observing the characteristics of the pedagogical domain.

This module shows and activates all anchors corresponding to links with a destination within the FNA and it only shows the anchors of those that go to areas different from the current one. In this way, the student is aware of which links can be seen and which cannot. The minimum number of areas available to the student is 1.

The possibility that the student could pass from one area to another is restricted to certain conditions which the system must evaluate and which basically consist in making sure that the student is familiar with the areas that were previously accessible.

Once conditions that allow access to an area have been verified, the system activates the anchors of the links that give access to it. Although it is generally considered that the hyperspace will increase in one by one areas, we do not restrict the possibility that at a certain moment the possibility of accessing two areas exists as in a branch.

The hyperspace in Figures 3, 4 and 5 will be used to describe the system operation. There are three different levels of accessibility: A1 (Figure 3), A2 (Figure 4) and A3 (Figure 5). At level A1 each one of the nodes is included in a different FNA. At level A3 there is only one area that includes all the nodes. Finally, level A2 is made up of four FNAs (those formed by the nodes AB, CD, EF and GH).

As previously indicated, it is necessary to define a partial ordering of navigation of FNAs. Whether the ordering is obtained automatically or not is irrelevant for the explanation of the module operation. In the three cases we will assume that the ordering determined for level of accessibility A1 is A-B-C-D-E-F-G-H. At level A2 the FNAs are composed by the nodes AB-CD-EF-GH and at level A3, there is obviously no ordering since there is only one FNA: ABCDEFGH.

It is also assumed that nodes A and B have already been visited (for this reason they are shaded in) and that the user is currently at node B.

For simplicity, it is also assumed that once the student visits a node, the node is considered visited at that moment and this is the condition set to give access to the following area(s). This means that in order to give a student access to node C on level A1 they must have visited the previous free surfing area (which contains node B) and we consider this to happen when node B is visited (and they would already know it). Similarly, in order to access free surfing area CD they must have first visited nodes A and B (and therefore they would have learnt the knowledge associated to the area AB).

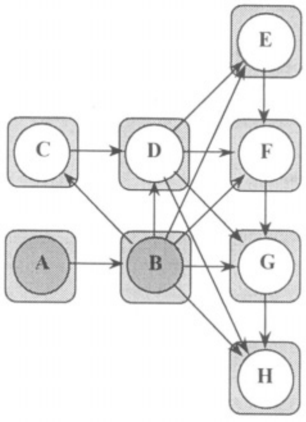


Figure 3. Level of accessibility A1

Table 2 shows the access possibilities between the different nodes on the different levels of accessibility, which are explained in the following paragraphs. Each link has two states: they can be available (A) or not (N).

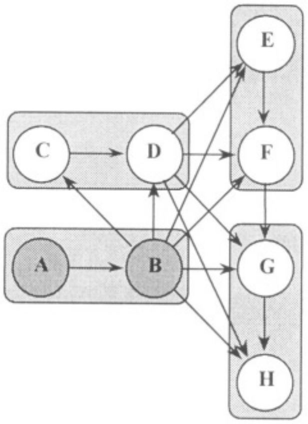


Figure 4. Level of accessibility A2

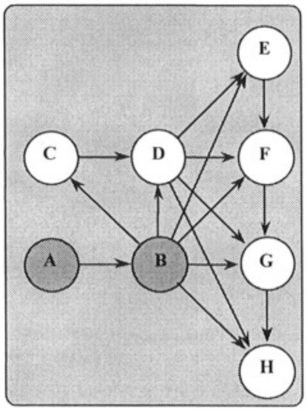


Figure 5. Level of accessibility A3

At level A1, the student can only access the nodes one by one. Therefore, when they arrive at node B all the links from that node will be available. However, they will only be able to cross those that lead to already visited nodes (A), those which lead to other nodes belonging to the available surfing areas at that moment (in this case only A) or links that the system decides can become visible (such as C).

Table 2. Availability of the links from node B according to the different levels of accessibility

	Type of access to the node						
Level	**A**	**C**	**D**	**E**	**F**	**G**	**H**
A1	A	A	N	N	N	N	N
A2	A	A	A	N	N	N	N
A3	A	A	A	A	A	A	A

At level A2, access is guaranteed to the nodes within the visible hyperdocuments (A, B, C and D, because AB have already been visited and the links to CD are activated once AB have been visited). If the hyperdocument EF were also visible, then access would be possible to the nodes A, B, C, D, E and F.

Finally, at level A3, the system enables direct access to all the nodes in the only FNA from the start, (which coincides with the hyperspace) provided that a link that leads to the nodes exists.

2.2 The levels of accessibility

The decision about the navigation ability to assign to a new student is one of the first tasks that should be undertaken and it is essential to assure the appropriate operation of the module. An incorrect decision on this level may make the student feel restricted by the scarce hyperspace available or overwhelmed (overloaded) by very extensive hyperspace, thus causing the student to reject the learning method.

Even though the objective of this module is to personalise the system's behaviour to the student's characteristics, at the beginning there is no data about the student's characteristics. We must remember that if the student has not worked with the system before it is advisable to use broader limitations at the beginning than those to determine the degree of system structuring, at least until the user is familiar with the system interface. Previous experience with the system (even in a different domain) may imply the use of wider free surfing areas because there is already one less obstacle to overcome (familiarisation with the system).

The number of accessible areas at the start usually depends on the student's knowledge of the domain presented in the hyperspace. All demonstrated knowledge has to redound in areas of the hyperspace where the student can navigate. However, only if the dexterity in the navigation is very low, the accessibility to smaller free surfing areas is recommended. The hyperspace will be widened relatively quickly, as the student becomes more familiar with its organisation so that in a short time all the knowledge demonstrated to be known is made available.

Anticipating that the levels of accessibility change throughout the course of the student's surfing, the module provides certain flexibility to adapt the levels of accessibility dynamically throughout a session. In this way, the risk that an incorrect initial decision will cause undesired effects is-reduced. The actual system carries out the adaptation after a request from the student's supervisor, and even if the actual user considers it necessary, from himself/herself. Certain: problems are associated with the level changes, which we will discuss later on.

The levels of accessibility can represent teaching strategies which are different from the domain. Therefore, different levels of accessibility can exist which do not correspond to equivalent abilities but correspond to the presentation of the domain from a different point of view. For example, a computer science course could be grouped according to the languages offered, the hardware used, or the operating systems, etc. All these ways of organising can be prepared for beginner students in computer science. Figure 6 shows this possibility graphically for the hyperspace shown, for example, in figure 3 assuming that there are only four levels of accessibility: A1, A21, A22 and A3. Level A1 is from Figure 3. Level A21 can be organised in four possible ways as shown in the lower part of Figure 6. Level A22 can be organised in two ways and finally we have level A3 from Figure 5.

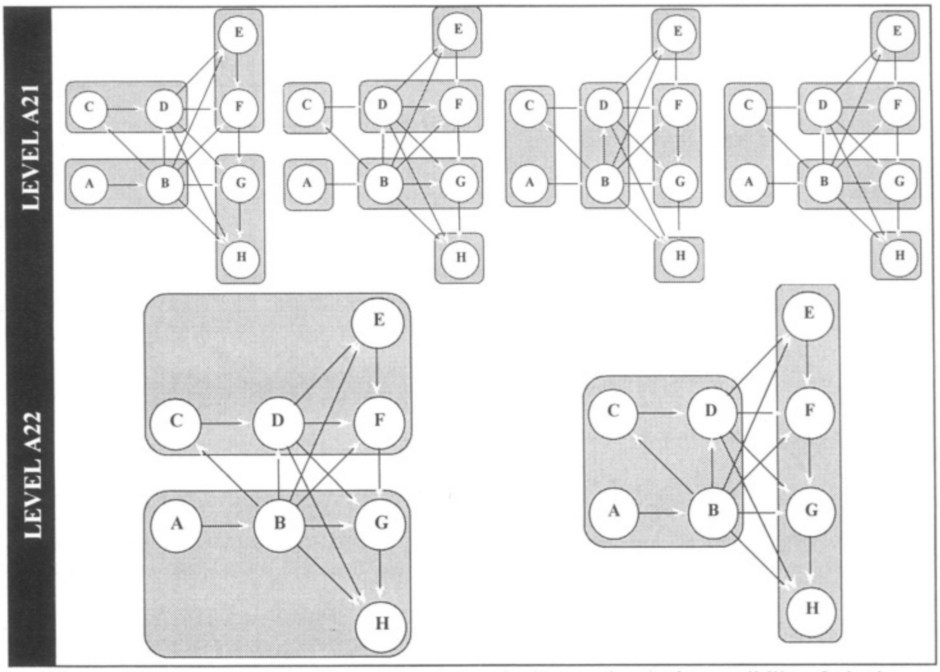


Figure 6. Different pedagogical points of view inside one level of accessibility. Only the intermediate levels appear in the figure. The previous and subsequent levels correspond to levels A1 and A3, respectively.

2.3 Changing the level of accessibility

The module allows changing the level of accessibility. It is recommended that the levels of accessibility to be changed use the same pedagogical strategy. The change can be carried out on the initiative of the student, the system or the student's supervisor.

When the student's level of accessibility is changed the hyperspace can change radically. The student has to be aware of this modification because the FNAs defined will change and this situation can be disorientating. The system is also aware of it and therefore controls the fact that the hyperspace visited by the student remains exactly the same as when the student visited it and in the nodes that have not been visited it operates as if they were at a new level. To do this, when a change has taken place the student is informed of it. The student will observe the evolution in the accessibility of the links. When the level of accessibility increases, more links will become available. Otherwise, the hyperspace will decrease and as expected some accessible links will become inaccessible.

The module classifies all the links of one node as available or not depending on the current area, the current node, the level of accessibility where the student is, and the actual destination of the link. They are given a label according to this classification and it is displayed next to the link. In this way, when the students build mental maps corresponding to the hyperspace, they have already seen all the links of one node.

The changes in the level of accessibility to wider areas do not cause any problems, since they only consist of making more links available. On the other hand if the areas decrease the change becomes more problematic. To clarify the system operation, let us assume that a user is at level A3 and at a given moment the access level is reduced to level A2. Let us also assume that on the new level, after an analysis of the student's knowledge, area CD is available but EF and GH are not. The system behaves differently depending on the current node and the relative position of the nodes visited against the areas available on the new level. Three different cases can be found which are described in the following sections.

2.3.1 Case 1: The nodes visited are a subset of the nodes in the accessible areas on the new level

The associated behaviour is to directly reduce accessibility, since there are no problems that impede it. Figure 7 shows a possible example of this case. Nodes A, B and D have been visited and the accessible areas on the new level are AB and CD, therefore the visible nodes are A, B, C and D, thus leaving areas EF and GH not available at the new level.

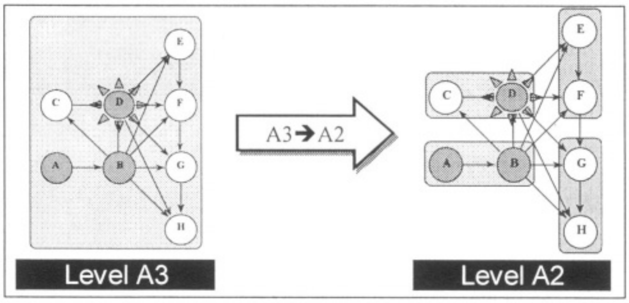


Figure 7. Evolution of the accessibility from level A3 to level A2 according to Case 1

2.3.2 Case 2: Visited nodes exist that belong to areas that are not accessible on the new level and the current node is inside the accessible areas on the new level

The system directly restricts accessibility (Figure 8). In this case the nodes visited are A, B and F. The current node is node B which is inside the accessible areas AB and CD. Access to areas EF and GH has been blocked (even to node F). The student must be informed of the level change and must be aware that access to node F has been closed because of the level change and that this does not have to affect the construction of any mental maps that were being created.

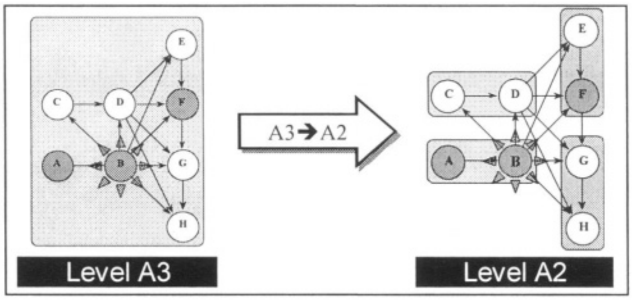
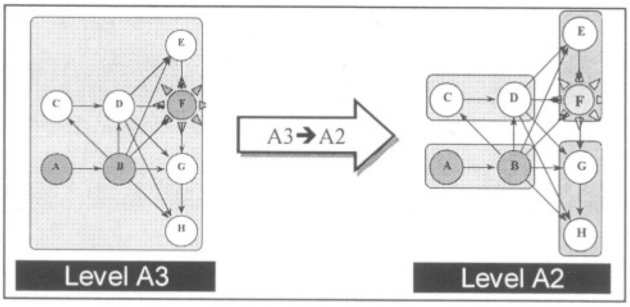


Figure 8. Evolution of the accessibility from level A3 to level A2 according to Case 2.

2.3.3 Case 3: Visited nodes exist and the current node belongs to non-accessible areas on the new level

In this case it is not possible to directly restrict the access since the user is inside a node that should be beyond their scope. Therefore the system makes all the nodes visited inside the area accessible while the student is inside it. As soon as the student leaves the hyperdocument the access to the hyperdocument will be prohibited until the necessary requirements are met so that it is opened on that level again.



Figures 9. Evolution of the accessibility of level A3 to level A2 according to Case 3.

This situation is not sufficient enough; the system must make sure that there is a path to the visible nodes on the new level. Otherwise, the system could leave the student isolated in a non-accessible area, as shown in Figure 10. G is the current node and after the level change, if area EF is left inaccessible then the student does not have the possibility to reach accessible areas AB and CD. The system must therefore leave the links to the nodes already visited in the inaccessible areas visible, so that the student can return to the permitted area of the hyperspace. The student will never be able to see new nodes inside the inaccessible areas.

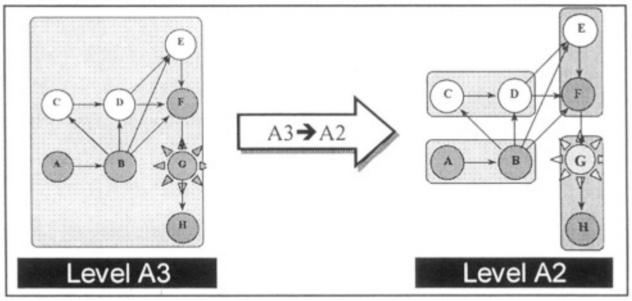


Figure 10. Problem that can appear in Case 3. The student loses all paths to return to the accessible areas.

3. CONCLUSIONS

In the previous sections we have described a method to limit access inside a hypermedia system for education so that the student's behaviour adapts to pedagogical strategies that could be designed. The method is very easy from the point of view of the author of the hyperspace because it is based on simple methods to contain the hyperspace nodes. Likewise, we have briefly looked at some of the problems that could arise with alternative hyperspace limitation methods and the way to overcome these problems with the proposed method. They are basically centred on not hindering the creation of student's mental maps of hyperspace, a problem that is reproduced when proposing a change on the level of accessibility during the student's navigation.

The adaptation of hyperspace provides instructive characteristics to hypermedia that enable the actual system to keep a certain part of the control that is yielded to the student when using hypermedia. However, the system has the necessary flexibility to be able to ignore these instructive characteristics if necessary. This enables this kind of systems to be used for instructive and constructive learning purposes. In the first case, more features are made available to the creator of the hyperspace.

The system proposed can avoid problems associated with hypermedia systems, such as disorientation, since it carefully avoids offering nodes that have different appearances (different links available) at different moments in time. Cognitive overload problems are also decreased because the number of links available to the student decreases. Likewise, the level of accessibility that is assigned to the student is variable so that it can adapt to their changing characteristics and increase the level of accessibility (the number of areas available) as the navigation ability increases and the organisation of the nodes that compose the working domain are learnt.

In any case, it must be said that this learning control system is based on the assumption that visiting a hypermedia node means that the contents it shows have been learnt, which may be very presumptuous. Therefore, we must say that the hypermedia learning system should be completed using elements that guarantee the learning acquisition and allow its transfer to new contexts to be checked. The HEZINET system incorporates an adaptive evaluation system based on the path covered, for example.

REFERENCES

1. Dick, W. (1991). An Instructional Designer's View of Constructivism. *Educational Technology* (5), 41-44.
2. Merrill, D. (1991). Constructivism and Instructional Design. *Educational Technology* (5), 45-52.
3. Nielsen, J. (1998). *Usability Engineering.* Academic Press: London.
4. Dede, C. & D. Palumbo. (1991). Implications of Hypermedia for Cognition and Communication. *International Association for Impact Assessment Bulletin,* 9 1-2, 15-28.
5. Pérez, T. A. (2000). *Un hiperentorno adaptativo para el aprendizaje constructivo/instructivo.* Tesis Doctoral, Universidad del País Vasco UPV-EHU, San Sebastián,.
6. López, R., J. Gutiérrez, A. González, J. A. Carro & T. A. Pérez. (1999). Cómo aprender euskara a distancia de forma adaptativa. *Actas del congreso nacional de informática educativa CONIED'99.* M. Ortega & J. Bravo, (eds.), Puertollano, Ciudad Real (Spain).
7. Fischer, P. M. & H. Mandl. (1990). Toward a Psychophysics of Hypermedia. In David H. Jonassen & Heinz Mandl (Eds.). *Designing Hypermedia for Learning.* NATO ASI Series, Springer Verlag: Berlin.
8. Ross, S. M. & G. R. Morrison. (1988). Adapting Instruction to Learner Performance and Background Variables. In D. H. Jonassen (ed.). *Instructional Designs for Microcomputer Courseware* Lawrence Erlbaum Associates Inc. Hillsdale, NJ (USA).

Improving the Language Mastery through Responsive Environments

A. Vaquero, F. Sáenz, A. Barco*
Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, E-28040 Madrid, Spain
** Computing Consultant, E-28040 Madrid, Spain*

Key words: computer based learning, constructionism, language mastery, electronic dictionaries, lexical database, ontology

Abstract: This paper is devoted to a way of intending to improve the language mastery of students in every subject and in a broad range of education levels by using computer-based tools involving linguistic resources. Lexicon concepts useful for language learning are involved in dictionaries and other kinds of linguistic resources. These concepts (vocabulary, meanings, semantic categories, semantic relationships, and taxonomy) are pointed out. Claim is made for using new environments and computer-human interfaces based on these concepts which define the pedagogical goals. Envisioning the foreseen collaborative instruction tasks, claim is made for the constructionist model of learning. Authoring and consulting electronic linguistic resources are the main tasks to reach the defined goals. The conceptual model appropriate for developing any software system taking into account all these principles is discussed, and the reached entity-relationship model is presented. We further present the developed tools for creating bilingual linguistic resources, which could allow reaching the foreseen learning goals. The authoring tool supports consistency of the intended semantics of the lexicon and enables the detection of omissions and inconsistencies, and the user tool is a graphical user interface for queries. These tools can be advantageously used in language teaching.

"Computers should improve learning and teaching into the classroom". Schneiderman [1]

1. INTRODUCTION

The challenge of integrating computers in education is mandatory but the traditional resistance of the Education World to innovation is well known. This is the general source of difficulties for exploiting the potentialities of New Technologies into Education [2]. Some very common facts observed in educational centres illustrate this assertion: lack of hardware in schools, lack of quality software, limitations of imported software, lack of reliable and trustworthy reviews or evaluations, etc.

The use of computers in the school must be controlled by the teacher [3] and so, he/she must be prepared for and aided in assuming this control. Besides the general Computer Literacy that every teacher must currently have, our "user teacher" should have a specific training in "Computers and Education" [4]. Particularly important is the aspect of Language, which we mention here because of its influence in the involved computer-based applications, not only in the direct communication between teacher and learner.

But generally, there is lack of support and appropriate training for teachers. Besides these difficulties, when teachers are committed to apply computers to the classroom, they detect a diversity of problems inhibiting the use of computers across the curriculum [5], some common ones are: quality of software, access to software, picking out useful software for their own teaching, much of the existing software is difficult to integrate into teaching, often teachers must put in more preparation time before the computer can be used in the classroom, etc. Moreover, as the learning approach changes, new problems appear. The changes are centred in the dynamics of learning, that is, the learner's processes for understanding. This new vision of the nature of learning is based on cognitive theories [6], [7] and it induces new ways of teaching [8]. Acquiring knowledge is an active construction process of the learner.

With this new vision in mind, the materials needed for teaching could be constructed thinking firstly in the learner more than in the curriculum [9]. As a consequence, new computer-based materials and tools should be created for aiding to the learner. Computer-based tools are necessary for off-load heavy, time consuming and boring activities: calculators, symbolic manipulators, graphic and statistical tools, computer-based laboratories, word processors, spread sheets, data bases, graphic editors, and so on. More important, the activities involved must induce motivation [10].

When weak domains in the student skills and knowledge are detected, it is mandatory to fill the gap by applying appropriate computer-based environments. An especially weak domain is Language. There exist a worrying lack in the mother tongue mastery of the young and not so young people, as it has been recently shown by reliable inquiries made in Spain, but more or less strongly the problem is felt all around the world. The key part of the language misunderstanding is lexicon. There is experimental evidence of reading comprehension dependency of the vocabulary [11, 12].

In order to improve the level of language mastery, every pupil ought to handle specific tools with facilities for creation, consulting and modification of language parts. The conventional dictionary, glossary, thesaurus, encyclopaedia, etc., electronic (mere electronic copy of the printed ones) or not, have been put in classroom for browsing and consulting purposes. It seems that electronic resources of this kind could motivate to the student more than paper ones. Firstly allowing to the student more easily and quickly to look up terms. Moreover, the computer could allow the development of new tasks with clear pedagogical goals according to the learning model based on constructionism [13]. This implies a new approach to the concept of electronic linguistic resources other than the electronic counterparts of the printed ones [14]. It is necessary to distinguish between machine-readable and machine-tractable dictionaries, as well as for every language resource.

The global pedagogical goal to be reached is word meaning [15]. Word definition is a task for intending to reach the learning of word meaning dependency on other words, as word classification is similar with respect to semantic categories. Specific goals are diverse relationships among terms such as polysemy and synonymy, and their implications into classification. There are involved concepts of the real world and relationships among them such as hypernymy and hyponymy closely attached to ontology [16]. The relations between two words of different languages are given through the ontology and its connection to both lexicons. All these goals can be reached following a constructive and collaborative way among students and the teacher in the classroom. This could only be efficiently done with appropriate tools and friendly usable interfaces as a whole responsive environment [17].

The intention is to improve the language mastery of students in every subject and in a broad range of education levels by using computer-based tools for authoring and consulting electronic linguistic resources. In order to situate the envisioned tools in their correct instructional place, one must distinguish between constructionist learning in user controlled environments (fully free environments) and navigation in hypermedia ones [18]. The envisioned tools belong to the first one of these two models of learning, the second one being more appropriate for learning other parts than lexicon [19]. Nonetheless, both are complementary and not absolutely separate [20]. As the second one has been extensively treated in the past [21], the first one is highlighted here. In this paper, we present computer-based tools for authoring and consulting multilingual lexical resources supporting ontologies. Our intention is to fill a gap in the niche of constructionist tools for improving the language mastery of students in every subject, which can be applied to a broad range of education levels. Moreover, we believe that each one must build his/her own personal dictionary along his/her entire life. Our tools aid to reach these goals. As far as we know, there are no similar tools to ours described in the literature.

This paper is organised as follows. In section 2, we highlight some linguistic concepts involved in multilingual dictionaries which are useful for language learning. In section 3, we present the conceptual model of the terminological database incorporated in our tools, which is illustrative for formalising the learning concepts of multilingual dictionaries. Section 4 presents the interface and functionalities of the software tools we have developed. Our first computer-based tool, a user tool for querying a bilingual dictionary, is presented in section 4.1. In section 4.2 we present our second computer-based tool, the author tool for creating bilingual dictionaries, which allows the accomplishment of more learning goals. Finally, section 5 summarises some conclusions and provides hints for future and related work.

2. LINGUISTIC CONCEPTS INCORPORATED IN OUR PROPOSAL

Linguistic concepts incorporated in computing systems devoted to Natural Language Processing are pointed out because of their relevance in the definition of the pedagogical goals. These systems subsume the conventional ones (paper dictionaries and the like). The links among these concepts and the pedagogical goals are highlighted. Lack of the kind of dictionaries we propose has been felt, as Fillmore et al. [22] states:

"... we imagine, for some distant future, an online lexical resource, which we can refer to as a 'frame-based' dictionary, which will be adequate to our aims. In such a dictionary (housed on a workstation with multiple windowing capabilities), individual word senses, relationships among the senses of the polysemous words, and relationships between (senses of) semantically related words will be linked with the cognitive structures (or 'frames'), knowledge of which is presupposed by the concepts encoded by the words."

In addition to the above need, a pedagogical goal is to reach a customisable personal dictionary that allows other functionalities such as personal notes, taxonomies, etc., which are not included in conventional dictionaries.

2.1 Order, Classification, and Ontology

Typically, monolingual dictionaries show an alphabetical order that can be seen as a simple term classification: terms are classified in singletons by its lexicographic form. Other possible less naïve classifications are derivative (root-shape), grammatical, and semantic. Derivative classifications [23] are not common, and grammatical classifications are not intended for dictionaries. Finally, semantic classification groups terms by semantic categories (for instance, synonym and antonym dictionaries, or ideological dictionaries [24].) Semantic categories not also allow meaning classification, but the more meaningful taxonomy of meanings. Conventional lexical databases, such as WordNet [25], have term classification such as synonymy (grouped in the so called synsets.) Ontologies go beyond by playing the role of meaning taxonomy [26]. Our tools do support this important concept as will be explained along the paper.

Semantic categories are useless for term lookups since meanings will correspond, in general, to a set of (synonym) terms[7]. However, it has an important role on learning by both using and authoring dictionaries because each meaning of a given term (polysemy and/or homonymy) is precisely identified by its semantic category (categories from now on, for the sake of brevity), instead of the usual nonsense sequential number[8]. Therefore, semantic categories provide classification for meanings, and such classification can be arranged in a taxonomy. But this not straightforwardly implies a term order since meanings are abstract ideas that cannot be expressed in general by one distinctive word[9]. It is commonly acknowledged that the best order for lookups is lexicographic (a derivative classification is a counterexample for this, but it still keeps a lexicographical order by repeating entries and adding links.) Figure 1 resumes the order for taxonomies in a hierarchy; it shows a taxonomy of categories along with the set of terms belonging to each category. From this point of view, there is a complete lexicographic order (provided categories are identified with terms or phrases.) A hierarchy is a natural structure for meaning classification. Each node in the hierarchy corresponds to a category. In principle, every category in the hierarchy can be used, no matter its hierarchy level. It must be noted that every category in the hierarchy contains at least the term which names the category, so that all categories are non-empty. On the other hand,

[7] Nevertheless, there are other kinds of term lookups as ideological dictionaries show.

[8] However, meaning identifications by numbers also show a coarse classification; e.g. Tech. for Technical.

[9] The question is: Which is the best word to represent a meaning? In general, there are several (synonym) words representing the same meaning.

the creation of new categories as intersection of several predefined ones should be avoided, in order to reach compactness.

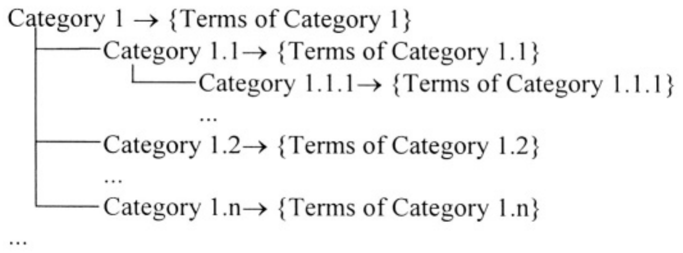

Category 1 → {Terms of Category 1}
Category 1.1→ {Terms of Category 1.1}
Category 1.1.1→ {Terms of Category 1.1.1}
...
Category 1.2→ {Terms of Category 1.2}
...
Category 1.n→ {Terms of Category 1.n}
...

Figure 1. A Taxonomy

3. EDUCATIONAL GOALS

From an educational point of view, the goal for students is not to develop a general dictionary (obviously, it is a huge work which linguistic researchers are still carrying out nowadays), but specialised and limited dictionaries that restrict the linguistic domain to make the categorisation of meanings and the definition of the taxonomy easier. There are a number of advantages in classifying meanings as a taxonomy. First, meaning taxonomy is a useful facility for an electronic dictionary because meanings embody additional semantics which provides more information to the reader (more than that of sequential numbers noted above.) Second, the system may also gain a new dimension because it is possible to automatically generate specialised dictionaries under different categories (a sports dictionary may deal with soccer, tennis, or baseball dictionaries.) Third, it helps to develop a balanced dictionary by adding enough terms from different categories. Having the terms classified, it is easy to check out how many terms are under a given category. Fourth, it also helps to distribute the work between several authors by assigning categories to authors. A team of authors may develop a complete specialised dictionary by dividing the work by categories so that collaborative work is promoted for students. This finally means that the categories must be defined, which implies an added bonus for educational purposes, since it means that students have to organise ideas in a formal way, supported by the implementation of the author tool (covered in the next section.) Additionally, meaning classification can be done under a grammatical criteria, that is, categories can refer to grammatical properties of words (nouns, verbs, ...), so that students can also learn grammatical aspects in this way.

3.1 Polysemy and Synonymy

In every language there exists the well-known naming problem, which consists of two elements: one is polysemy (under the synchronic point of view, that is, embodying polysemy itself and homonymy), by which a term can have several meanings; and the other is synonymy, by which one meaning can be assigned to different terms, as can be observed in Figure 2. In this Figure, Term 1 and Term 2 are synonyms and have a shared meaning, as so for Term 2 and Term3, under another meaning. Moreover, Term 2 is polysemic since it has two possible meanings.

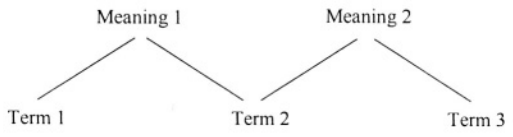


Figure 2. Polysemy and Synonymy

3.2 Relationships

Here we do some remarks about the relationships between categories, meanings and terms. On the one hand, a given term can belong to several categories under different meanings. On the other hand, a given term can belong to several categories under the same meaning. Figure 2.3 shows two categories (C1 and C2) which respectively contain the meanings {M11, M12, M} and {M, M21, M22}. Each meaning has one or more terms associated. The term T2 is associated to meanings M12 and M21, which respectively belong to categories C1 and C2. We also show the term T that is assigned to meaning M, which belongs to both categories C1 and C2. Polysemy is present in T2, and synonymy is also present in T3, and T4, as it can be seen. T1 is neither polysemic nor synonym. TC1 and TC2 are the terms used to denote categories C1 and C2, respectively.

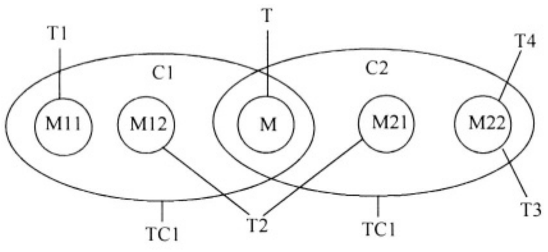


Figure 3. Relationships among categories, meanings and terms. Extensional definition

In this figure, the set of meanings {M11, M12, M} in C1 is the extensional definition of category C1. We must also note that a category has a meaning described by a definition. This figure does not embody this fact. In order to embody the meanings related to categories, we transform the scheme of Figure 3 to the one depicted in Figure 2.4. Now, C1 is the meaning of the category C1, and TC1 is the term assigned to such meaning, and the same applies to C2 and TC2. Then, we have one more meaning in each category. This meaning is the intensional definition of the category.

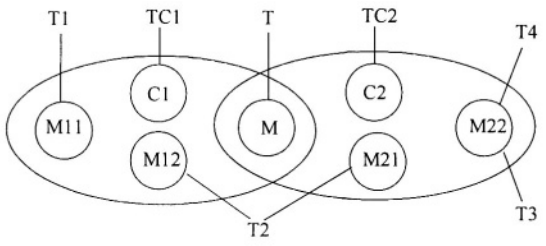


Figure 4. Relationships among categories, meanings and terms. Intensional definition

For a given language, we have a set of terms that holds the relationships with categories and meanings shown in Figure 2.4. If we now think of several languages, the same applies for each one. Then, relationships between terms from different languages come from considering jointly the involved schemes.

3.3 Foreign Language Learning

Besides the advantages noted above in using and authoring electronic dictionaries with ontologies for language learning, we also consider foreign language learning under, first, the same lexicon meaning point of view, and second, the relationships between the foreign language and the mother tongue, that is, a multilingual electronic dictionary with support for ontologies. The key idea here is to develop an environment which embodies the linguistic concepts noted above for both the mother tongue and one or more foreign languages. This environment will be a multilingual electronic dictionary with ontologies, and it can allow reaching several teaching goals. First, the assimilation of the foreign lexicon in a constructive way. Second, the mastery of the foreign language when defining meanings in the foreign language. Third, if grammatical criteria are applied to classification, then foreign grammatical properties of the foreign language can be exerted. Finally, the student can comprehend the language independence notion of meanings, so that semantic categories can independently be defined from the language.

4. CONCEPTUAL MODEL OF THE TERMINOLOGICAL DATABASE (TDB)

There are different TDBs built for different purposes. Some of them have incorporated the ontology structure and, so they could possibly be used for the pedagogical goals proposed above. But there are a lot of difficulties when intending to do this, not being the less the fact that these very large databases are yet complete or almost complete. So only the tools for building terminological databases are needed. Moreover, the development of this kind of tools must be made taking into account the pedagogical goals which have not been the case of the LDB already built.

Our work in developing the tools is based on a sound conceptual model for the terminological database which shall eventually hold the terms, definitions, meanings, and semantic categories. Since it is intended to deal with two or more languages (bilingual or multilingual dictionaries), we need to represent instances of terms, textual definitions, and textual semantic categories for each language, but, as meanings are not language dependent, we shall use unique representations for them.

The entity-relationship model is used to describe the conceptual model we propose shown in Figure 5. In this Figure (following some recommendations in [27] and [28]), entity sets are represented with rectangles, attributes with ellipses, and relationship sets with directed and undirected lines. If B has an incoming line from A, this denotes a one (A) to many (B) mapping cardinality. Double arrows denote many to many mapping cardinalities. Undirected lines denote both one to one mapping cardinalities, when connecting entities, and attributes of entities, when connecting attributes to entities. Relationship set names (not shown in this Figure) label each line.

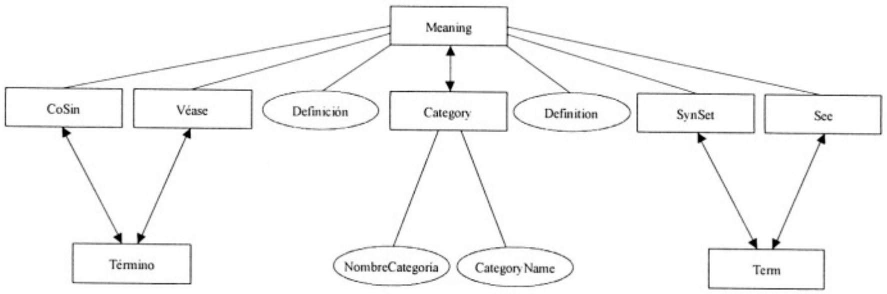


Figure 5. Entity-Relationship Model for an English-Spanish TDB

For the sake of clarity and conciseness, in this Figure we show an instance of a multilingual terminological data base for only Spanish and English languages, but it naturally derives from the general model depicted in Figure 7 (where Li denotes the i-th language, $i \in \{1,..,N\}$.) In Figure 5 we depict the entity Meaning, the central entity other entities rest on. In fact, this is the entity which is language independent. The entity SynSet denotes the English synonym set (SynSet - Synonym Set.) The relationship set between both entities is one to one. The entity Term represents all the English terms that compose the terminological database. The relationship set between SynSet and Term is many to many since a synonym set contains several terms, and a term may be contained in several synonym sets (obviously, with different meanings). Figure 6 embodies this idea, in which Term 1 and Term 2 are synonyms and has a shared meaning, as so for Term 2 and Term 3, under another meaning. Moreover, Term 2 is polysemic.

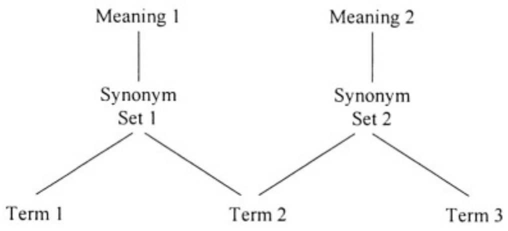


Figure 6. Polysemy and Synonymy related with the synonym sets

The entity See denotes the set of English terms related under a given meaning. The relationship set between Meaning and See is one to one. The relationship set between See and Term is many to many, because a meaning may refer to several English terms, and one term may be polysemic. The entity Category denotes the category each meaning belongs to. The relationship set between Category and Meaning is many to many since many meanings are in a category, and a meaning could be in several categories (this situation is expected to be reduced to the minimum since the goal is to keep the classification as disjoint as possible). This relationship set embodies the fact that our classification is not lexical (there is not a direct relationship between Category and Term) but semantic (we relate meanings to categories, i.e., we categorise meanings.) The entity Category has two attributes: CategoryName and NombreCategoría, which correspond to the textual name of the category in each considered language, English and Spanish, respectively. Meaning has two attributes: Definition and Definición, which correspond to the textual definition in the same considered languages. The remaining entities (CoSin, Véase, Término) are homologous to the respective entities (SynSet, See, Term).

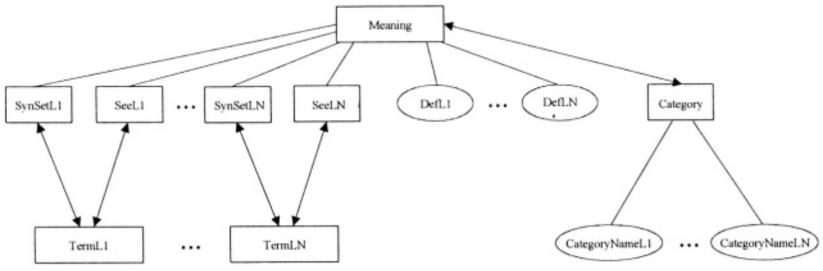


Figure 7. Entity-Relationship Model for a Multilingual TDB

The logical and physical models for the development of any terminological database following the principles above expressed have to be based on this conceptual model.

5. INTERFACE AND FUNCTIONALITIES OF THE TOOLS

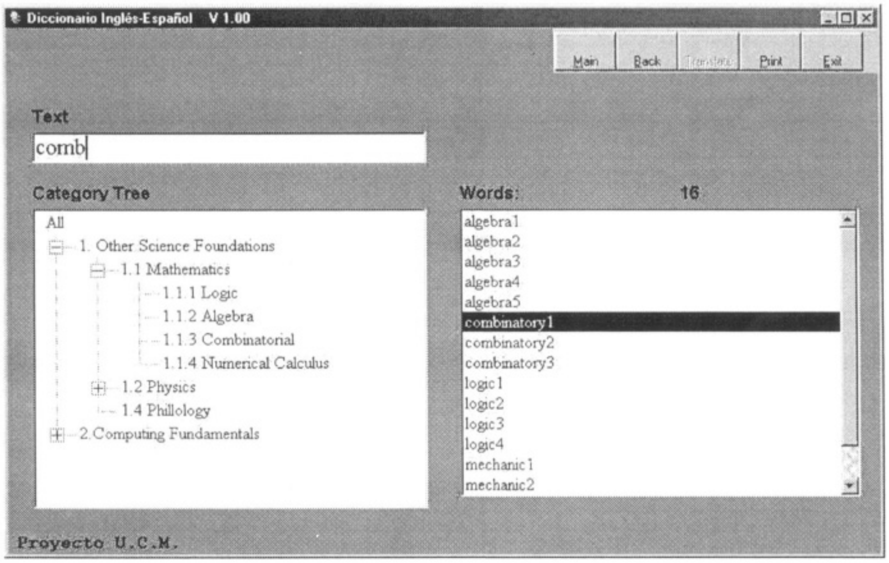
This section briefly describes the interfaces and functionalities of the user tool and the author tool we have developed.

5.1 The User Tool

We have developed a user tool, a query interface which allows us to easily recover the information about both English and Spanish terms as well as their relationships from the so-called terminological database. This database holds the terms, categories, their attributes, and the relationships. The interface allows the user to navigate the semantic categories, also allowing retrieving the relevant information of any term (definition, other related terms, translation, synonyms, ...).

The Start window of this tool allows the user to select the base language (i.e., the source language for translations and for representing dialogues) among the available languages by pressing its button (from now on, we consider a bilingual dictionary so that it is unnecessary to select the source language or the target language).

This action pops up the Semantic Category window, as shown in Figure 8; its left pane shows the semantic categories structured as a tree, and the right pane, all the words under the highlighted semantic category. The total number of terms is showed on top of the right pane. The nodes in the tree can be clicked in order to expand or contract semantic categories subtrees. A text box is used for term lookups so that the closest word to the substring typed is shown in the right pane. Pressing Enter or double-clicking the highlighted word yields to the Query window. This window shows the relevant information about the selected term: its definition, comments, the list of semantic categories it belongs to (the one corresponding to the shown definition is highlighted), the synonym set and the list of related terms. It also displays a navigation history. It is possible to select another semantic category in this window, which results in updating all the relevant information. Direct access to the terms in both the synonym and related terms windows is allowed by double-clicking.



Text: comb

Figure 8. Semantic Category Window

The Semantic Category window has a control box with buttons to activate the return to the Start window, navigate backwards, translate the selected word, print, and exit the interface. The Translate button offers one of the main functionalities of this interface, i.e., the translation from the (source) base language to the target language and, when pushed, it pops up the Translation window (Figure 9). This window shows a first field for the term in the first language, and a second field for the term in the second language. There are also navigation buttons for searching other terms in the same semantic category under an alphabetical order. It is possible to translate from the first or from the second language by using two buttons which express the two possible translation directions. Also, the Go to buttons allow us to go to the Semantic Category window for the selected term. This completes the overall description of the functionalities of the user tool.

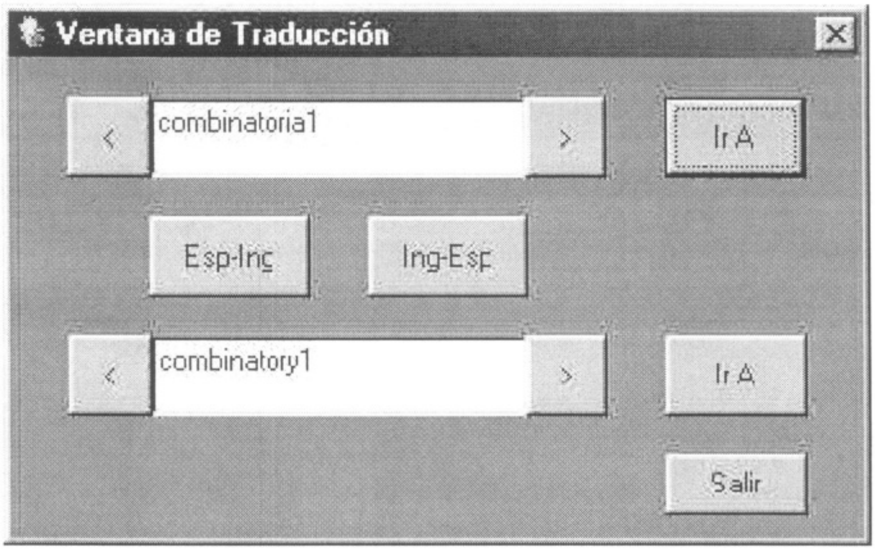


Figure 9. Translation Window

5.2 The Author Tool

The author tool allows the author to add new terms to the terminological database, and all the relevant information, such as its definition, semantic categories, meanings, synonym sets, and related terms. We have developed a Spanish user interface for this tool (easily rewritable to other language), and it consists mainly of one Author window, as shown in Figure 10. It has several management areas (indicated by superimposition in this figure) which are explained next.

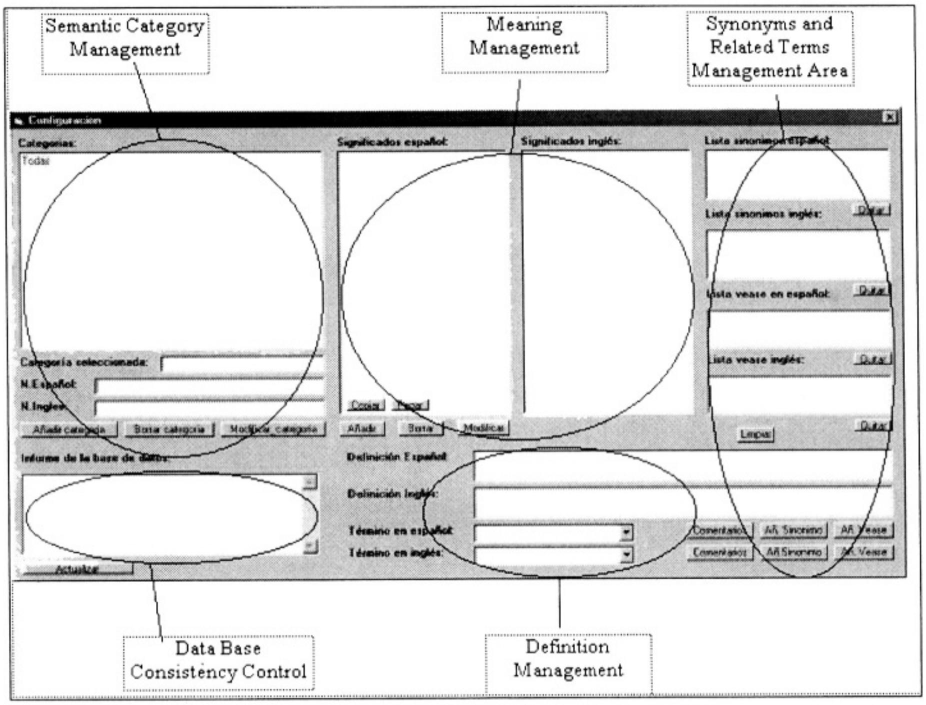
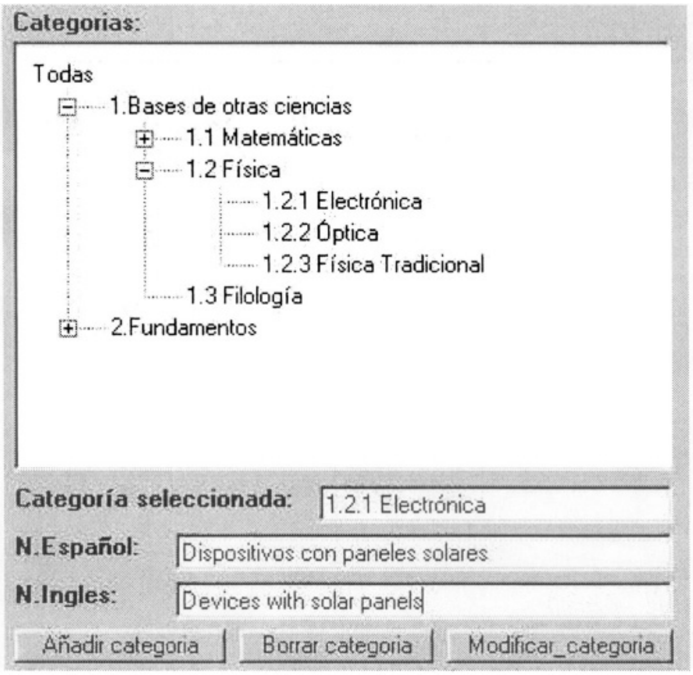


Figure 10. Author Window

5.2.1 Semantic Category Management

This area is intended for managing all the operations related to semantic categories, as illustrated in Figure 11 with a fragment of a taxonomy. It has several controls: a hierarchical view of the semantic categories (with expand/collapse functionality), text fields for the semantic category names (English and Spanish), and the buttons Add Category, Delete Category, and Modify Category. The insertion point when adding a new semantic category is the highlighted semantic category, and the Spanish and English texts for the semantic category name must be typed in the aforementioned text fields.



Categoría seleccionada: 1.2.1 Electrónica

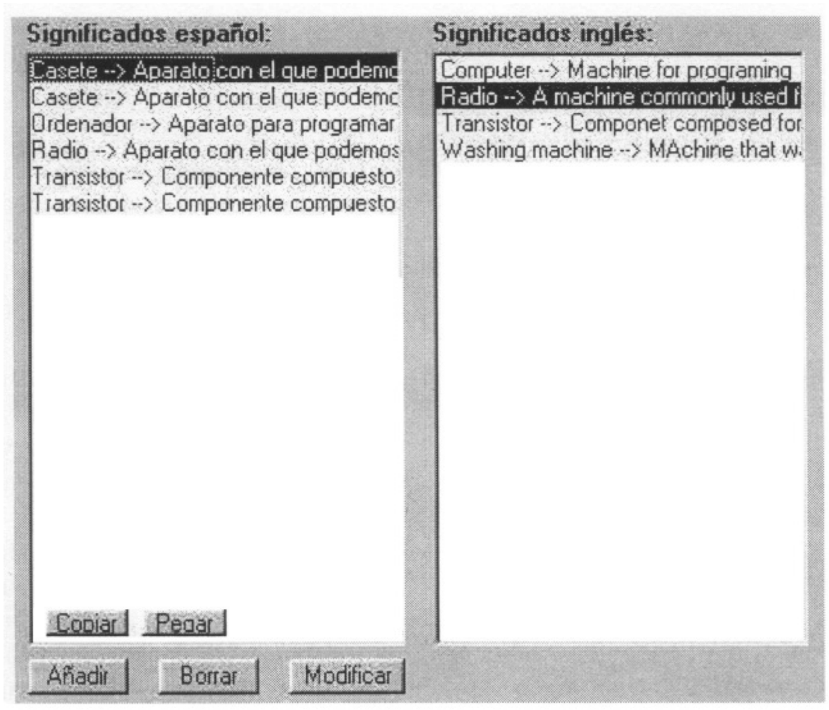
N.Español: Dispositivos con paneles solares

N.Ingles: Devices with solar panels

Figure 11. Semantic Category Management Area

5.2.2 Meaning Management

The area for meaning management, illustrated in Figure 12, consists of two lists for the meanings in both languages and the buttons Add, Delete, and Modify for addition, deletion, and modification of meanings, as well as buttons for edition (Copy and Paste buttons.) These lists shows the meanings in the form Term -> Definition for the highlighted category, so that one can see several meanings for the same term. Moreover, when a pair Term -> Definition is selected, the corresponding Term -> Definition translation is automatically highlighted; there is a one-to-one mapping between meaning representation in all the languages. It should also be noted that meanings, which are language independent, are shown with the *best* representation we have in a given language, i.e., a pair Term -> Definition, since there are no other pair Term -> Definition2 with the same meaning (note that is the same term in both pairs.)



Casete --> Aparato con el que podemc
Casete --> Aparato con el que podemc
Ordenador --> Aparato para programar
Radio --> Aparato con el que podemos
Transistor --> Componente compuesto
Transistor --> Componente compuesto

Computer --> Machine for programing
Radio --> A machine commonly used f
Transistor --> Componet composed for
Washing machine --> MAchine that w.

Figure 12. Meaning Management Area

5.2.3 Synonyms and Related Terms Management Area

The area on the right in Figure 12 has four lists for the synonyms, and related terms in both languages which correspond to the highlighted meaning in the Meaning Management area.

5.2.4 Database Control Area

This area contains the button Update, which is used to modify the database with the typed information, and to obtain a report (text box Data Base Report) about consistency of the database (Figure 3.3). Up to now, consistency detection only detects lack of textual definitions for terms, but it can be extended in order to detect other inconsistencies or omissions. This is quite important when authoring dictionaries, since a dictionary cannot be consistently built at each step, but it is constructively built from terms to relationships between terms (polysemy, synonymy.) For instance, this tool can be extended in order to give hints for detecting circular definitions (there are commercial dictionaries with this failure), for detecting possible lacks of synonym and related terms, and so on.

6. CONCLUSIONS

To face the challenge of fulfilling the lack of language mastery new tools and interfaces are needed. These tools have to be a consequence of the linguistic concepts involved, the defined pedagogical goals, and the chosen constructionist learning model, as exposed above. So, a sound entity-relationship model which embodies these linguistic concepts has been established, and it is well suited for developing lexical databases with a rich structure. The conceptual model presented here supports the concept of ontology.

We have developed tools for aiding to improve the language learning. These tools are a consequence of the defined pedagogical goals and the linguistic concepts involved, as shown above. Putting to work in service these represents a hope for educational achievement, and the experiences will guide the additions and modifications to improve their efficacy.

These tools can be enhanced in several ways. To mention only a few, firstly, both the user and author tool can be deployed in a Web context in order to allow centralised information for queries, and, what is more important, to allow collaborative work at a distance. Secondly, they can be extended with phonetic search. And thirdly, the author tool database control can be improved with the identification of not defined words in textual definitions, which can help for completeness.

Classification of meanings, as we have emphasised before, is important for two challenging applications. First, to integrate a terminological database into a multilingual knowledge base. And second, for information retrieval. A sound conceptual model is necessary to integrate a terminological database into a multilingual lexical one. We have in mind lexical knowledge bases based on ontologies (e.g., MikroKosmos [29]), rather than monolingual on line lexical resources (e.g., WordNet [25].) Since the conceptual model we develop here is coherent with the concept of ontology, the implementation of a terminological database from that conceptual model must facilitate its integration into an ontology based lexical knowledge base. So we can assume the implementations to be built from this model will accomplish the conditions to be rightly used as essential steps of information retrieval operations.

REFERENCES

1. B. Schneiderman, M. Alavi, K. Norman, and E.Y. Borkowski, "Windows of opportunity in electronic classroom", Communications of the ACM, vol 38, n 11, November, 1995.

2. B. Cornu, ”New Technologies: Integration into Education”. Proceedings of the Working Conference”, Integrating Information Technology into Education”, IFIP, Barcelona, pp.15-24, October 17-21, 1994.
3. L. Cuban, “Teachers and Machines: The classroom use of technology since 1920”. Teachers College Press, Columbia University, New York, 1987.
4. F.J. Erickson, and J.A. Yonk, ”Computer Essentials in Education. The teaching tools”. McGraw-Hill Book Co., 1994.
5. B. Hodgson, ”The roles and the needs of the teacher”. Proceedings of the Working Conference ”Integrating Information Technology into Education”, IFIP, Barcelona, pp. 25-34, October 17-21, 1994.
6. D.P. Ausubel, “Educational Psychology: A cognitive View”, New York, Holt, Reinhart and Winston, 1968.
7. M.I. Posner(Ed.), ”Foundations of Cognitive Science”. Cambridge, Mass., MIT press, 1989.
8. K. Tobin, and D. Tippings, ”Constructivism as a referent for teaching and learning”. K. Tobin (Ed.), ”The practice of Constructivism in Science Education”. AAAS Press. Washington, DC, 1993, pp. 3-21.
9. J. Karat, “Evolving the scope of user-centered design” Communications of the ACM, Vol. 40, N. 7, July, 1997.
10. L. Cuban, Foreword in “Teaching with Technology: Creating Student Centered Classrooms”. H.J. Sandholtz, C. Rigstaff, and D.C. Dwyer, Eds., Teachers College Press, New York, 1997.
11. D.D. Johnson, and P.D. Pearson, ”Teaching Reading Vocabulary”, Ed. Holt, Reinhard & Winston, New York, 1978.
12. R.L. Thorndike, ”Reading Comprehension Education in Fifteen Countries”, Ed. Wiley, 1973.
13. A. Cabrera, “Informática Educativa: La revolución constructivista”. Informática y Automática, Vol. 28, n. 1, marzo 1995.
14. Y.A. Wilks, D.C. Pass, C.M. Guo, J.E. McDonald, T. Plate, and B.M.Slator, ”Providing machine tractable dictionary tools”. Machine Translation, 5, 1990, pp. 99-151.
15. M. R. Quillian, “Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities”. Brachmen, R. J. y Levesque, H. J., Eds., Reading in Knowledge Representation. Morgan Kaufman, 1.967.
16. B. Onyshkevych, and S. Nirenburg, “Lexicon, Ontology, and Text Meaning”, Lexical Semantics and Knowledge Representation, J. Pustejovsky, Ed., Springer-Verlag, 1992.
17. D. Zeltzen and R. K. Addison, “Responsive virtual environments”, Communications of the ACM, Vol. 40. N. 8, August, 1997.
18. K. Norman, “Navigating the educational space with HyperCourseware”. Hypermedia, Vol. 6, enero 1994.
19. S.R. Goldman, ”Reading,Writing,and Learning in Hypermedia Environments”, Cognitive Aspects of Electronic Text Processing (Ed.H. Van Oostendorp and S. Mul), Norwood, NJ. Ablex Publications, 1996.
20. P. Teusch, T. Chanier, Y. Chevalier, D. Perrin, F. Mangenot, J.P. Narcy, and J.de Saint Ferjeux, ”Environnements interactives pour l’apprentissage en langue étrangère”. Hipermedias et Apprentissage, 3 (Ed. E. Brouillard), 1996, pp. 247-256.
21. A. Fernández-Valmayor, C. López-Alonso, S. Arlette, and B. Fernández-Manjón, “The Design of a Flexible Hypermedia System: Integrating an Interactive Learning Paradigm for Foreign Language Text Comprehension”, International Working Conference on Building Electronic Educational Environments, IFIP, Irvine, California, pp. 51-65, 1999.

22.C.J. Fillmore, and B.T. Atkins, “Toward a frame-based lexicon: The semantics of RISK and its neighbors”, Lehrer and Kittay, pp. 75-102, 1992.
23. M. Moliner, Derivative Spanish Dictionary.
24.Casares, Ideological Spanish Dictionary.
25. G. Miller, “WordNet: A Lexical Data Base for English”, Communications of the ACM, Vol. 38, 11, 1995.
26. S. Nirenburg, V. Raskin, and B. Onyshkevich, “Apologiae Ontologiae”, Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, Center for Computational Linguistics, Catholic University, Leuven, Belgium, pp. 106-114, 1995.
27.R.S. Pressman, "Software Engineering. A Practitioner's Approach", McGraw-Hill, 1997.
28.A. Silberschatz, H.F. Korth, S. Sudarshan, "DataBase System Concepts", WCB/McGraw-Hill, 1996.
29. MikroKosmos, http://crl.nmsu.edu/Research/Projects/mikro/index.html

Index